



UNIVERSIDADE NOVA DE LISBOA
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica

UMA INFRAESTRUTURA COMPUTACIONAL DE GESTÃO DE DADOS NUM
CONTEXTO DE EFICIÊNCIA ENERGÉTICA

Por:
António Victor Segão Nunes

Dissertação apresentada na Faculdade de Ciências e Tecnologia
da Universidade Nova de Lisboa para a obtenção do grau
de Mestre em Engenharia Electrotécnica e de Computadores

Orientador: Doutor Celson Pantoja Lima
Co-Orientador: Doutor João Francisco Alves Martins

Monte de Caparica
2011

Copyright

UMA INFRAESTRUTURA COMPUTACIONAL DE GESTÃO DE DADOS NUM
CONTEXTO DE EFICIÊNCIA ENERGÉTICA

António Victor Segão Nunes - Todos os direitos reservados.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa tem o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Dedico à minha família..

Agradecimentos

Em primeiro lugar e acima de tudo aos meus pais, António Nunes e Ana Carolina Nunes pela educação que me deram e por todo o apoio demonstrado, incutiram em mim a ambição e determinação necessária para terminar o curso, que por sua vez possibilitou a escrita deste documento. Ao meu irmão Luís Nunes e à minha tia Domingas Segão pelo apoio manifestado desde sempre em todas as etapas da vida e do curso.

À minha companheira Cláudia Leal pelo amor, carinho e paciência em mim depositados que me ajudou bastante durante a escrita desta dissertação.

Aos restantes membros da minha família que sempre me incentivaram a prosseguir até a esta meta académica que me lança no mundo do trabalho.

Gostaria também de agradecer ao meu orientador Doutor Celson Lima que me guiou de uma forma eficiente durante todo o processo de elaboração da dissertação, permitiu-me adquirir métodos eficazes de organização, pesquisa e implementação de ideias, atributos fundamentais para uma boa evolução na vida profissional que me espera.

Ao meu co-orientador Doutor João Martins pela, não menos importante, ajuda na elaboração da dissertação e aos restantes membros participantes, professores e alunos, do projecto *NEMO*.

Ao meu grande amigo José Oliveira e novamente à minha tia Domingas Segão pelas verificações e correcções feitas no processo de escrita deste documento.

A todos os meus amigos e companheiros de curso, que estiveram sempre a meu lado durante a caminhada até esta etapa. O tempo que passei em companhia de pessoas excelentes contribuiu imensamente para o meu crescimento pessoal e profissional, graças ao companheirismo de todos.

Quero mais uma vez manifestar o meu sincero agradecimento a todos aqueles que contribuíram, de uma forma ou de outra, para o meu sucesso no decorrer do curso e da dissertação. Para todos vocês, um grande Obrigado.

Resumo

A eficiência energética é um tema bastante discutido na actualidade. O desenvolvimento económico mundial nas últimas décadas, tem conduzido a uma utilização intensa de energia produzida a partir de recursos de origem fóssil (carvão, petróleo, gás natural, etc.). Estes recursos têm várias desvantagens: não são renováveis, as instalações que os exploram são frágeis e têm uma pegada ecológica enorme, relativamente à sua produção, distribuição e consumo. Estas desvantagens apontam para uma necessidade de mudança do modelo de desenvolvimento económico. Nesta perspectiva, surgiu no final do século XX o conceito de desenvolvimento sustentável, que tem em conta o equilíbrio ecológico e a qualidade de vida das populações. Existe assim necessidade de utilizar mecanismos com custos financeiros menores, que permitam poupar e gerir os recursos existentes com o uso de energias renováveis para uma eficiência energética maior.

A visão global desta dissertação assenta no facto de uma rede de *Intelligent Electronic Device* (IED) relacionados com energia podem ser operados com a ajuda de uma infra-estrutura (distribuída) de software, onde existe a necessidade de configuração dos dispositivos, armazenamento e gestão das informações dos dispositivos e visualização de dados para posterior administração.

Este trabalho visa a criação de uma infra-estrutura de *software* constituída por três partes: a inteligência, os mecanismos de acesso ou interface e a gestão dos dados, no contexto de um sistema mais amplo voltado para a eficiência energética.

Esta infra-estrutura intitulada como *NEMO Manager* (NEMO-M), armazena e gere as informações referentes aos IED que se encontram na sua rede. A informação alusiva aos dispositivos encontra-se formatada pelo *standard International Electrotechnical Commission's* (IEC) 61850, através do uso da *Abstract Communication Service Interface* (ACSI) e da *Substation Configuration Description Language* (SCL), que possibilita interoperabilidade e escalabilidade do *software*. O NEMO-M disponibiliza ainda um portal *web* que serve como interface para os utilizadores, que lhes possibilita o acesso às distintas informações contidas nas diferentes bases de dados dos sistema.

Palavras Chave: Eficiência Energética, Energia, Energias Renováveis, Portal Web, Serviços Web, IEC 61850.

Abstract

Energy efficiency is currently a major issue. The world economic development in recent decades has led to an intense energy production from fossil resources (coal, oil, natural gas, etc.). Fossil resources have several drawbacks: they are non-renewable; facilities that exploit them are fragile; and they have a large ecological footprint regarding their production, distribution and consumption. The aforementioned disadvantages point to a need of change of the economic development model. From this perspective emerged in the late twentieth century, the concept of a sustainable development model that accounts the ecological balance and the population's quality of life.

From the aforementioned context, there is a need to use mechanisms, with lower financial costs, that save and manage existing resources with renewable energy sources to provide better energy efficiency.

This dissertation is based by the fact that a IED related network can be operated helped by an software distributed infrastucture, where is a need of device configuration, storage and management of information of the devices and future visualization of the data for administration.

This work aims to create a software infrastructure composed of three parts: the intelligence, the mechanisms for access or interface, and data management in the context of a larger system aimed at energy efficiency.

This infrastructure called NEMO-M, stores and manages the information of all *Intelligent Electronic Device* (IED) that are on its own network. The information regarding the device is based on the standard *International Electrotechnical Commission's* (IEC) 61850, through the use of *Abstract Communication Service Interface* (ACSI) and the *Substation Configuration Description Language* (SCL), which enables the interoperability and scalability of the software.

Keywords: Energy Efficiency, Energy, Renewable Energy, Web Portal, Web Services, IEC 61850.

Conteúdo

Copyright	iii
Agradecimentos	vii
Resumo	ix
Abstract	xi
Acrónimos	xxi
1 Introdução	1
1.1 Enquadramento do Problema	1
1.2 Objectivos da Dissertação	2
1.3 Visão Geral da Dissertação	2
1.4 Contexto de Desenvolvimento da Dissertação	3
1.5 Estrutura do Documento	4
2 Sistemas de Gestão de Energia	5
2.1 Energy Management Systems	5
2.1.1 Visão Geral dos Sistemas SCADA	6
2.1.1.1 Arquitectura dos Sistemas SCADA	6
2.1.1.2 Evolução dos Sistemas SCADA	8
2.1.1.3 O <i>Hardware</i> usado nos Sistemas SCADA	8
2.1.1.4 Conceitos Fundamentais dos Sistemas SCADA	9
2.1.2 Algumas Tecnologias relacionadas com EMS e sistemas SCADA	10
2.1.2.1 Uma Implementação para Controlo e Monitorização de Sistemas EMS/SCADA	10
2.1.2.2 O Futuro dos Sistemas <i>Supervisory Control and Data Acquisition</i> (SCADA) para Sistemas de Distribuição Descentralizados	11
2.1.2.3 Um Perfil para Electrodomésticos baseado nos EMS	13
2.1.2.4 Sistema de Supervisão e de Gestão de Energia para Grandes Edifícios Públicos	15
2.2 Síntese e Análise	16
3 Escalabilidade e Interoperabilidade nos Sistemas de Gestão de Energia	19
3.1 Interoperabilidade	19
3.1.1 <i>Service-Oriented Architecture</i>	20

3.1.2	<i>Web Services</i>	21
3.1.2.1	Processo de Contratação de um <i>Web Service</i>	21
3.1.3	Importância do Uso de <i>Standards</i> na Indústria ICT	22
3.1.4	<i>Standards</i> Usados no Trabalho Desenvolvido	24
3.1.4.1	<i>International Electrotechnical Commission's 61850</i>	24
3.1.4.2	<i>Substation Configuration Description Language</i>	25
3.1.4.3	<i>Abstract Communication Service Interface</i>	26
3.1.4.4	<i>eXtensible Markup Language</i>	28
3.2	Escalabilidade	29
3.3	Síntese e Análise	29
4	Solução Proposta	31
4.1	Requisitos do Projecto	31
4.2	Solução Proposta: Visão Funcional, Visão Estrutural	33
4.2.1	Visão Funcional	33
4.2.2	Visão Arquitectural	36
4.3	Constituintes do NEMO-K	39
4.3.1	NEMO Service Manager (Internal Services)	40
4.3.2	IED Configuration Manager	40
4.3.3	Setup / Configuration	41
4.4	Diagrama de Entidades e Relações (DER)	41
4.4.1	Base de Dados <i>IED</i>	41
4.4.2	Base de Dados <i>Services</i>	43
4.4.3	Base de Dados <i>Users</i>	44
5	Implementação	49
5.1	Escolhas Tecnológicas Feitas no Decorrer do Projecto	49
5.2	Visão Estrutural	50
5.2.1	Diagramas de Classes	50
5.2.1.1	Camada de Interface do NEMO-M	51
5.2.1.2	Camada de Controlo do NEMO-M	51
5.2.1.3	Camada de Entidade do NEMO-M	56
5.2.2	Diagramas de Sequência	56
5.2.2.1	Funcionalidade <i>Login</i>	59
5.2.2.2	Funcionalidade <i>AcceptUser</i>	60
5.2.2.3	Funcionalidade <i>Inserção de um IED</i>	60
5.2.2.4	Funcionalidade <i>Inserção dos Serviços</i>	61
5.2.3	Diagramas de Actividade	62
5.3	Exemplo de Utilização	64
6	Validação	67
6.1	<i>IED Configuration Manager</i>	67
6.1.1	Exemplo em Funcionamento	67
6.2	<i>NEMO Service Manager (Internal Services)</i>	69

6.2.1 Exemplo em Funcionamento	70
6.3 <i>NEMO External Services</i>	71
7 Conclusões	73
7.1 Síntese Geral do Trabalho	73
7.2 Conclusões	74
7.3 Trabalho Futuro	74
Bibliografia	77

Lista de Figuras

1.1	Redes no Sistema NEMO. Adaptado de [Lima et al., 2011a]	4
2.1	Implementação Proposta Baseada em <i>Web Services</i> e nos <i>Standards</i> IEC [Mercurio et al., 2009].	10
2.2	Diagrama de Blocos que representam um DG. Adaptado de [Geberslassie and Bitzer, 2010].	12
2.3	Arquitectura de alto-nível do DEHEMS [Chao et al., 2010].	14
2.4	Arquitectura do Sistema DEHEMS [Chao et al., 2010].	15
3.1	Processo Geral de Contratação de um <i>Web Service</i> . Adaptado de [WS0, 2004].	22
3.2	Estrutura Esquemática da Arquitectura SCL [Huo et al., 2008].	27
3.3	Virtualização de um IED usando ACS I [Lima et al., 2011b].	27
4.1	Os Pilares do Sistema NEMO.	32
4.2	Vista do Sistema NEMO	33
4.3	Diagrama Casos de Uso de um Utilizador	34
4.4	Diagrama Casos de Uso do Administrador	35
4.5	Relações ICE do NEMO-M	36
4.6	Vista Geral do NEMO Kernel	39
4.7	Diagrama de Entidade e Relação da base de dados <i>IED</i> - Componente SCL Heading	42
4.8	Diagrama de Entidade e Relação da base de dados <i>IED</i> - Componente NEMO File	43
4.9	Diagrama de Entidade e Relação da base de dados <i>IED</i> - Componente SCL	45
4.10	Diagrama de Entidade e Relação da base de dados <i>IED</i> - Componente <i>Data Type Templates</i>	46
4.11	Diagrama de Entidade e Relação da base de dados <i>Services</i>	47
4.12	Diagrama de Entidade e Relação da base de dados <i>Users</i>	47
5.1	Diagrama da Classe <i>RegisterForm</i>	51
5.2	Diagrama da Classe <i>RegisterAction</i>	52
5.3	Diagrama da Classe Principal desta Componente	52
5.4	Diagramas das Classes <i>Parsers</i>	53
5.5	Diagrama da Classe <i>NemoFile</i>	54
5.6	Diagrama da Classe Principal desta Componente	54
5.7	Diagrama da Classe do tipo <i>IED</i>	55
5.8	Diagrama da Classe do tipo <i>Identification</i>	55
5.9	Diagrama da Classe do tipo <i>Device</i>	56
5.10	Diagrama da Classe <i>Inserts</i>	57

5.11 Diagrama da Classe <i>QuerysIEDs</i>	58
5.12 Diagrama da Classe <i>Inserts</i>	58
5.13 Diagrama da Classe <i>QuerysServices</i>	58
5.14 Diagrama da Classe <i>ServicesUtils</i>	58
5.15 Diagrama da Classe <i>Utils</i>	59
5.16 Diagrama de Sequência da funcionalidade <i>Login</i>	59
5.17 Diagrama de Sequência da funcionalidade <i>AcceptUser</i>	60
5.18 Diagrama de Sequência da funcionalidade <i>Inserção dum IED</i>	61
5.19 Diagrama de Sequência da funcionalidade <i>Manage Services</i>	61
5.20 Diagrama de Actividades da funcionalidade <i>Login</i>	63
5.21 Página Inicial do NEMO-M	64
5.22 Dispositivos na Rede do NEMO-M	64
5.23 Dispositivo Novo na Rede do NEMO-M	65
5.24 Estado do Dispositivo Alterado	66
6.1 Funcionamento do módulo <i>IED Configuration Manager</i>	67
6.2 Primeira <i>Query</i> à base de dados dos IEDs.	68
6.3 Excerto do <i>SCL File</i> do " <i>Aerogerador</i> ".	68
6.4 Excerto do <i>NEMO-File</i> do " <i>Aerogerador</i> ".	69
6.5 Segunda <i>Query</i> à base de dados dos IEDs.	69
6.6 Funcionamento do módulo <i>NEMO Service Manager (Internal Services)</i>	70
6.7 Primeira <i>Query</i> à base de dados dos <i>Services</i>	70
6.8 Segunda <i>Query</i> à base de dados dos <i>Services</i>	71
6.9 Métodos do <i>Web Service</i> que interage com a base de dados dos <i>Services</i>	72
6.10 Métodos do <i>Web Service</i> que interage com a base de dados do <i>Users</i>	72

Lista de Tabelas

2.1	Comparação entre os Sistemas Descentralizados e os Centralizados [Geberslassie and Bitzer, 2010].	13
4.1	Exemplos de Relações entre as três Camadas - Utilizador	37
4.2	Exemplos de Relações entre as três Camadas - Administrador	38

Acrónimos

ACSI *Abstract Communication Service Interface*

API *Application Programming Interface*

AVAC *Aquecimento, Ventilação e Ar Condicionado*

BAS *Building Automation Systems*

BEMS *Building Energy Management Systems*

CID *Configured IED Description*

CIM *Common Information Model*

DA *Data Attribute*

DER *Diagrama de Entidades e Relações*

DG *Distribution Generation*

DMS *Distribution Management Systems*

DEHEMS *Digital Environment Home Energy Management System*

DO *Data Object*

DPWS *Devices Profile for Web Services*

EMCS *Energy Management Control Systems*

EMS *Energy Management Systems*

EPS *Electrical Power Systems*

GID *Generic Interface Definition*

HMI *Human-Machine Interface*

HTML *HyperText Markup Language*

HTTP *HyperText Transfer Protocol*

ICE *Interface, Controlo e Entidade*

ICT *Information and Communications Technology*

IDE *Integrated Development Environment*

IEC *International Electrotechnical Commission's*

IED *Intelligent Electronic Device*

IP *Internet Protocol*

IT *Information Technology*

JSP *JavaServer Pages*

LD *Logical Device*

LN *Logical Node*

MTU *Master Terminal Unit*

MVC *Model-View-Controller*

NEMO *NEtworked MOnitoring and COntrol, Diagnostic for Electrical Distribution*

NEMO-C *NEMO Connector*

NEMO-F *NEMO File*

NEMO-K *NEMO Kernel*

NEMO-M *NEMO Manager*

NEMO-P *NEMO Portal*

P2P *Peer-to-Peer*

PAC *Programmable Automation Controller*

PLC *Programmable Logic Controller*

PnP *Plug and Play*

RES *Renewable Energy Sources*

RTU *Remote Terminal Units*

SAS *Substation Automation System*

SCADA *Supervisory Control and Data Acquisition*

SCL *Substation Configuration Description Language*

SGML *Standard Generalized Markup Language*

SOA *Service-Oriented Architecture*

SOAP *Simple Object Access Protocol*

SQL *Structured Query Language*

ORM *Object/Relational Mapping*

UML *Unified Modeling Language*

WPF *Windows Presentation Foundation*

WSDL *Web Services Description Language*

XML *eXtensible Markup Language*

XSD *XML Schema Definition*

Capítulo 1

Introdução

Neste capítulo é apresentado e descrito o tema do trabalho, as suas qualidades e história. Em seguida, é explicada a relevância do tema no contexto actual e a motivação que sustenta o projecto.

1.1 Enquadramento do Problema

A eficiência energética é um dos grandes temas que se discute na actualidade. O desenvolvimento económico mundial nas últimas décadas, tem conduzido a uma utilização intensa de energia produzida a partir de recursos de origem fóssil (carvão, petróleo, gás natural, etc.). O carácter não renovável desses recursos, a vulnerabilidade das instalações para a sua exploração, processamento e distribuição e o impacto ambiental da sua produção e consumo, apontam a necessidade de mudança de modelo de desenvolvimento. Nesta perspectiva, surgiu no final do século XX o conceito de desenvolvimento sustentável, em resultado do reconhecimento de que o desenvolvimento económico deve ter em linha de conta o equilíbrio ecológico e a preservação da qualidade de vida das populações.

Como é do conhecimento geral, a energia é na actualidade, uma das principais preocupações. Tanto os ambientes residenciais, como empresariais, são cada vez mais e mais dependentes de energia. O aumento acentuado da procura por uma fonte de energia, tornou-se quase num síndrome que é testemunhada por exemplo nos aeroportos de todo o mundo, onde os lugares mais desejados são aqueles mais próximos de uma fonte de corrente [Lima et al., 2011a]. É também um reflexo de que todos necessitamos de energia para as actividades básicas, tais como o trabalho e o lazer.

A crise de energia destacou dois protagonistas que estão interligados: a eficiência energética e as fontes de energia renováveis. Existe portanto, uma crescente necessidade de eficiência quanto ao uso de energia com vista a evitar o desperdício da mesma, como tentar que seja o mais sustentável ou ecologicamente correcto o seu uso. A eficiência energética tornou-se então, neste momento, um dos pilares da sustentabilidade global. Neste contexto, *Information and Communications Technology* (ICT) é uma grande mais-valia, sendo um instrumento capaz de ajudar a catalisar a harmonização adequada desses dois actores [Lima et al., 2011a].

1.2 Objectivos da Dissertação

O foco de desenvolvimento do trabalho descrito por esta dissertação, consiste em explorar a próxima geração de dispositivos de rede, os IED, de modo a suportar a monitorização dinâmica de dispositivos inteligentes, como por exemplo medidores de energia, fontes de alimentação, electrodomésticos e equipamentos industriais.

Este trabalho visa a criação de uma infra-estrutura de *software*, de seu nome *NEMO Manager* (NEMO-M), constituída por três partes: os mecanismos de acesso ou interface, a gestão dos dados e a inteligência, no contexto de um sistema mais amplo voltado para a eficiência energética.

Para a evolução positiva do *software* será necessário o desenvolvimento duma interface que possibilite que um administrador/utilizador usufrua do sistema, neste caso um portal *web* que com o auxílio de diversos *Web Services* fará a ligação entre o portal e o núcleo do NEMO-M.

O núcleo do NEMO-M será composto por dois módulos de inteligência. Um módulo que permita tratar e organizar a informação contida em dois ficheiros, guardando a mesma numa base de dados para o efeito. Um outro que aceda à base de dados referida filtre e organize a informação necessária e por sua vez proceda à construção completa de uma nova base de dados.

Elaborar e administrar as diferentes bases de dados que permitam o bom funcionamento do *NEMO-M* torna-se então um factor indispensável. Para tal serão necessárias a criação de três bases de dados: primeira, a base de dados que guarda as informações sobre os diferentes IEDs que estão na rede do sistema; segunda, a base de dados que guarda as informações sobre os serviços que estão disponíveis na rede e qual o dispositivo que o disponibiliza e uma terceira, a que guarda as informações sobre os utilizadores do sistema.

1.3 Visão Geral da Dissertação

Com os problemas actuais já referidos, derivados da escassez dos recursos fósseis, verificou-se uma viragem para o recurso às energias chamadas limpas, ou renováveis. Esta mudança de recurso energético advém também de uma maior consciencialização do homem em relação ao meio ambiente e das contrapartidas que surgem do intenso uso da energia produzida a partir de recursos não renováveis, tais como o carvão, o petróleo, o gás natural, etc.. Estes factores puseram assim o tema da eficiência energética na ordem do dia. É então do interesse geral da população, a evolução dos mecanismos de consumo, de produção e de distribuição da energia com o intuito de poupança da mesma, que contribuem para a preservação dos recursos naturais existentes, assim como para baixar os níveis de poluição no planeta.

Será de esperar que o NEMO-M se traduza numa ferramenta útil, no que respeita ao auxílio quanto à eficiência energética em sistemas, ou conjunto de sistemas, onde exista consumo, produção e distribuição de energia.

A visão global desta dissertação assenta no facto de uma rede de IED relacionados com energia podem ser operados com a ajuda de uma infra-estrutura (distribuída) de *software*, onde existe a necessidade de configuração dos dispositivos, armazenamento e gestão das informações dos dispositivos e visualização de dados relevantes sobre os mesmos para posterior administração.

Com esta visão torna-se necessário cumprir os objectivos descritos na secção 1.2, deste modo, será possível uma diminuição da energia consumida, porque tudo será efectuado de uma forma mais racional e ambientalmente correcta, devido à monitorização e ao controlo inteligente que o *NEMO-M* irá possuir.

1.4 Contexto de Desenvolvimento da Dissertação

A dissertação encontra-se englobada num projecto intitulado como *NEtworked MOnitoring and COntrol, Diagnostic for Electrical Distribution*, financiado pelo Quadro de Referência Estratégico Nacional (QREN), em parceria com as empresas portuguesas Critical Software e ATECNIC e também com a empresa internacional Schneider Electric.

A visão conceptual que guia o *NEtworked MOnitoring and COntrol, Diagnostic for Electrical Distribution* (NEMO), considera que redes de dispositivos relacionados com energia podem ser operados com a ajuda de uma infra-estrutura de *software* distribuído, suportada pelo paradigma *Service-Oriented Architecture* (SOA) e baseada em *standards*. Assim, qualquer sistema chamado de sistema NEMO é composto por duas redes, nomeadamente: a rede de energia e a rede de *software*, representadas na Figura 1.1. A primeira é formada por sistemas e dispositivos de produção, distribuição e consumo de energia. A última é usada para administrar e controlar a rede de energia [Lima et al., 2011a].

O projecto NEMO visa o desenvolvimento de uma infra-estrutura de *software* baseada em *standards*, com o objectivo de prestar o apoio necessário para realizar a gestão e controlo de dispositivos relacionados com a energia (por exemplo, fontes de energia renovável, medidores de energia, relés, etc.), considerando um ambiente onde a energia é gerada, armazenada, distribuída e consumida de uma forma racional e ambientalmente correcta.

A contribuição desta dissertação no projecto NEMO, consiste no desenvolvimento de uma infra-estrutura de *software* para a gestão de dados do projecto, bem como a elaboração de um portal *Web* para acesso ao mesmo. Posto isto, falta ainda referir que esta dissertação assenta somente na rede de *software*.

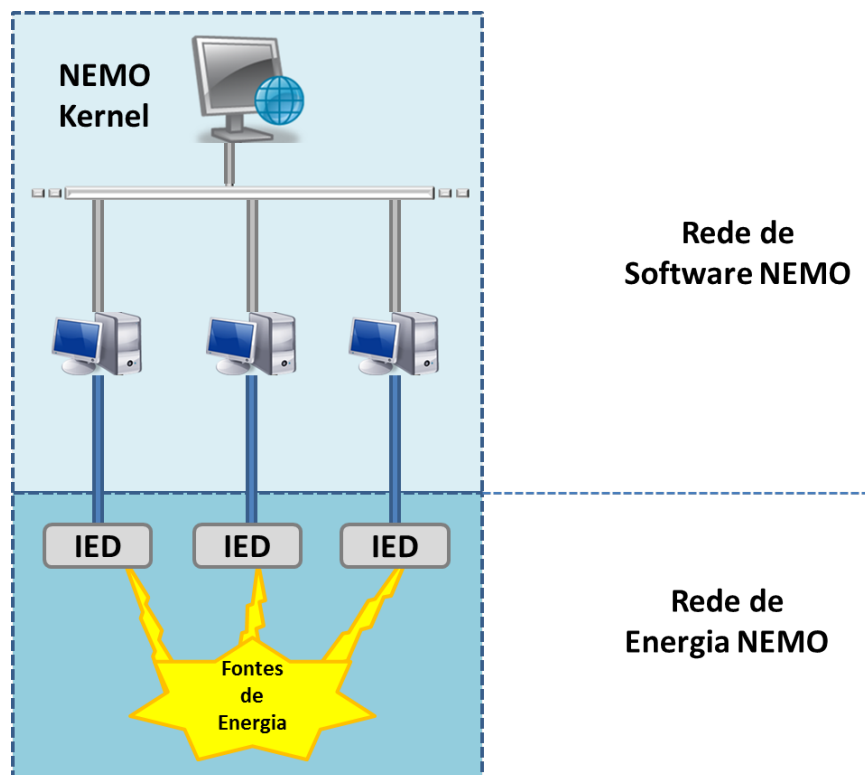


Figura 1.1: Redes no Sistema NEMO. Adaptado de [Lima et al., 2011a]

1.5 Estrutura do Documento

Esta dissertação está estruturada em sete capítulos, descritos nos parágrafos seguintes.

O Capítulo 2 aborda o contexto do trabalho realizado, nomeadamente os sistemas que servem como base do mesmo.

O Capítulo 3 analisa a escalabilidade e a interoperabilidade dos sistemas de gestão de energia, num ponto de vista dos *standards* usados e qual a sua importância no trabalho efectuado.

O Capítulo 4 visa dar conhecimento ao leitor sobre sistema desenvolvido, através de uma visão de alto nível sobre a sua estrutura e o seu funcionamento.

O Capítulo 5 versa sobre a implementação efectuada na dissertação, abordando temas como as tecnologias utilizadas e, por fim, um exemplo de utilização.

O Capítulo 6 ilustra o funcionamento do trabalho elaborado no decorrer desta dissertação.

O Capítulo 7, o das conclusões, está reservado para apresentar os resultados alcançados, bem como indicar possíveis linhas de continuação deste trabalho.

Capítulo 2

Sistemas de Gestão de Energia

Este capítulo tem o intuito de elucidar o leitor sobre áreas de suporte ao desenvolvimento do trabalho realizado, falando sobre os *Energy Management Systems* (EMS) que englobam, entre outros, os sistemas SCADA.

2.1 Energy Management Systems

A evolução no controlo automático de edifícios começou na década de 80, tendo adquirido cada vez mais importância na área do controlo de energia, o que levou a que se tornasse nos dias de hoje uma prática normal, isto é, praticamente todos os edifícios não residenciais, têm controladores automáticos com um computador como processador central. Estes sistemas são denominados por *Energy Management Systems*, *Energy Management Control Systems* (EMCS) ou *Building Automation Systems* (BAS).

A tecnologia de controlo está constantemente a evoluir, o que levou com que os proprietários dos edifícios e administradores das instalações tenham cada vez mais em conta a verificação sobre as melhorias a implementar nos mesmos. De salientar, que até para EMS recentemente instalados, há inúmeras possibilidades para substituição do sistema, ou apenas actualização, como por exemplo: computadores mais potentes, mais controle de nível de zona, sensores mais precisos, melhores serviços, etc.. O avanço contínuo da tecnologia, juntamente com a natureza dinâmica dos edifícios actuais, torna essas decisões de avanço tecnológico cada vez mais complicadas para os proprietários dos edifícios.

Muitas das funcionalidades avançadas existentes nos EMS são mal aproveitadas. Por exemplo, as capacidades de propensão e de monitorização dos EMS, são ferramentas poderosas para melhorar o Aquecimento, Ventilação e Ar Condicionado (AVAC), a iluminação e a redução da utilização de energia desnecessária. Este subaproveitamento deriva do facto dos responsáveis pela compra, ou actualização do EMS, na maioria dos casos, confiarem nos vendedores que fornecem as especificações sobre o sistema e acabam por não estudar a fundo as potencialidades que realmente os sistemas oferecem, podendo ser o caso de se ter instalado um sistema que não é o mais indicado para o edifício em

questão [Stum et al., 1997].

2.1.1 Visão Geral dos Sistemas SCADA

Os sistemas SCADA são geralmente associados ao sistema de controlo da indústria e são sistemas computacionais que controlam e administram diferentes processos. Estes processos podem ocorrer numa infra-estrutura, numa instalação ou numa zona industrial, tais como [SCA, 2011]:

- Os processos de infra-estrutura podem ser privados ou públicos, e incluem estações de tratamento de água e distribuição, recolha e tratamento dos resíduos da água, distribuição e transferência de energia eléctrica, gasodutos e oleodutos e ainda os sistemas de comunicação de grande porte.
- Estações espaciais, aeroportos, navios e as instalações de edifícios públicos ou privados, têm todos processos específicos de instalação. Estas instalações são de diversos tipos, como por exemplo, o controlo e a gestão do acesso aos edifícios, o seu consumo, o Aquecimento, Ventilação e Ar Condicionado e a sua energia.
- Os processos industriais referidos incluem: a produção; refinação; fabricação e geração de energia e podem ser executados em lotes, de modo contínuo, discreto ou repetitivo.

2.1.1.1 Arquitectura dos Sistemas SCADA

Um sistema SCADA é normalmente constituído pelos seguintes elementos:

- Um dispositivo que serve de interface entre o sistema e o operador humano mostrando todos os dados processados, auxiliando ainda o operador a controlar e a monitorizar os processos. Este dispositivo é chamado *Human-Machine Interface* (HMI).
- Um sistema de supervisão que processa todos os dados necessários sobre o processo.
- *Remote Terminal Units* (RTU), ou em português, Unidades de Terminais Remotas, que estão conectados aos sensores, recolhendo informação que enviam para o sistema de supervisão.
- *Programmable Logic Controller* (PLC), quando são usados como dispositivos de campo ao invés dos RTU, porque são mais versáteis, com maior capacidade de configuração, mais económicos e flexíveis.
- Infra-estrutura de comunicação que conecta os RTU ao sistema de supervisão.

HMI

O HMI é um aparelho que apresenta os dados processados ao operador humano e com o qual, o processo é controlado pelo mesmo. O HMI está ligado às bases de dados do sistema de modo a

possibilitar a troca de informações com o especialista do sistema, como por exemplo informações lógicas, esquemas detalhados para uma determinada máquina ou sensor ou ainda procedimentos de manutenção e de guias de solução de problemas.

As informações facultadas pelo HMI aos operadores são geralmente gráficas, sob a forma de diagramas. Esta representação possibilita mostrar ao operador a planta do sistema que está a ser controlado sob a forma duma representação esquemática. Por exemplo, a imagem de uma bomba ligada a um cano, mostra ao operador que esta bomba está a funcionar e a quantidade actual de líquido que percorre a tubagem e permite que a bomba seja actuada, isto é, possibilita que o operador desligue ou ligue a bomba a qualquer momento.

Estação de Supervisão

Uma Estação de Supervisão é formada por um sistema computacional e os respectivos servidores que são responsáveis pela comunicação com os equipamentos de campo, como os PLCs, os RTUs, entre outros, seguindo para o *software* HMI que está a funcionar na sala de controlo, ou em outro lugar.

A *Master Station* de um sistema SCADA pode ser composta por vários servidores, por locais para a recuperação de falhas e por aplicações de *software* distribuído nos sistemas SCADA de grande porte.

PLC

Um *Programmable Logic Controller* é um dispositivo que foi criado para substituir os circuitos de relé sequencial que eram necessários para controlar uma máquina. O PLC funciona verificando constantemente os seus *inputs* e consoante o seu estado, actuar nos *outputs* pretendidos.

Um PLC é uma forma especial dum controlador baseado num micro-processador que utiliza memória programável para armazenar instruções e para implementar funções, tais como lógica, sequências, temporização, contagens e aritmética, de modo a controlar máquinas e processos. Foi desenvolvido de modo a permitir que engenheiros com poucos conhecimentos em computadores e em linguagens de programação os pudessem operar.

RTU

O RTU está ligado no equipamento físico e converte, muitas vezes, todos os sinais eléctricos do equipamento para valores digitais, como por exemplo o estado do mesmo para aberto/fechado, através de uma válvula ou um *switch*, altera ainda algumas medidas como o fluxo, a pressão, a corrente ou a tensão do mesmo. Com essa conversão e com o envio do sinal eléctrico para o equipamento, o RTU pode controlar equipamentos como fechar ou abrir uma válvula ou um *switch*, ou para ajustar a

velocidade de uma bomba de água [SCA, 2011].

2.1.1.2 Evolução dos Sistemas SCADA

- **Primeira Geração: Monolítico.**

A computação na primeira geração foi feita com a ajuda dos sistemas *Mainframe*. Quando o sistema SCADA foi desenvolvido ainda não havia redes de computadores, portanto os sistemas operavam sem qualquer ligação a outro sistema, sendo por isso sistemas independentes. Mais tarde, os vendedores de RTUs projectaram as *WAN*¹ que ajudaram na comunicação com o RTU. Em caso de alguma falha ou avaria que levasse a que o sistema *Mainframe* principal falhasse, existia um sistema *Mainframe* de reserva que estava ligado ao nível do modo *Bus*. Este facto tornou a primeira geração de sistemas SCADA pouco viável, pois era redundante.

- **Segunda Geração: Distribuído.**

A informação partilhada entre as várias estações era feita em tempo real através de *LAN*² e o processamento de dados era distribuído entre várias estações. O custo e tamanho das estações usadas reduziu em comparação com as estações usadas na primeira geração pelo facto da responsabilidade por uma tarefa ter sido atribuída apenas a uma estação.

- **Terceira Geração: Rede.**

É esta geração que é usada actualmente nos sistemas SCADA, e onde o sistema em vez de usar um ambiente proprietário, que é controlado pelo vendedor, optou por usar um sistema de arquitectura livre para o utilizador. Permitindo assim a conectividade de qualquer dispositivo periférico ao sistema e onde as comunicações feitas com a estação principal são feitas através do protocolo *Internet Protocol* (IP) [SCA, 2011].

2.1.1.3 O Hardware usado nos Sistemas SCADA

Algumas soluções dos sistemas SCADA muitas vezes possuem os componentes de controlo distribuídos. A execução de processos lógicos de fácil resolução, sem ter que envolver o computador principal, está a aumentar dia após dia devido ao uso de PLCs ou RTUs inteligentes. O *standard* IEC61131-3 ou *Ladder Logic* como é mais conhecido, sendo uma linguagem de programação gráfica com foco em blocos passou a ser frequentemente usada na criação de programas que correm em PLCs ou RTUs. Esta linguagem de programação tem uma grande vantagem, devido ao facto de não necessitar de formação específica, precisando apenas de alguns requisitos de formação, ao contrário das linguagens de processos como FORTRAN ou linguagens de programação como C. Este facto permitiu que os engenheiros de sistemas de SCADA pudessem implementar e desenhar os programas de modo a

¹Wide Area Network é uma rede de computadores que abrange grandes áreas.

²Local Area Network é uma rede de computadores que liga computadores e dispositivos num espaço limitado, tal como uma casa.

serem executados nos PLCs ou nos RTUs.

No *hardware* dos sistemas SCADA existe ainda um controlador compacto, o *Programmable Automation Controller* (PAC), que combina as capacidades e as características do sistema de controlo baseado nos computadores, com qualquer PLC normal. De modo a fornecer funções dos PLC e dos RTU, os PACs são posições nos sistemas SCADA. Existe ainda a possibilidade de colocar RTUs distribuídos nas subestações eléctricas em diversas aplicações SCADA, computadores usados nas estações ou processadores de informação para a comunicação com os PACs, relés de protecção e outros dispositivos I/O, de modo a serem usados em vez dos tradicionais RTUs que são utilizados para comunicar com o SCADA Master [SCA, 2011].

2.1.1.4 Conceitos Fundamentais dos Sistemas SCADA

Os sistemas SCADA dizem respeito aos sistemas centralizados que controlam e monitorizam por inteiro algum local, ou então, são a parte complexa dum sistema espalhado em grandes áreas, considerando como grandes áreas, espaços que vão desde uma planta industrial a um País e onde todas as acções de controlo são maioritariamente feitas automaticamente pelos RTUs ou pelos PLCs. As restrições existentes para as funções de controlo do sistema por parte do *host* do mesmo, estão ao nível da supervisão ou ajustes básicos. Por exemplo: o PLC num processo industrial controla o fluxo de refrigeração da água, o sistema SCADA permite aos operadores a possibilidade de alterar as condições para o qual o sistema alerta com um alarme e ainda ajustar os pontos de controlo do fluxo, como a temperatura ou a perda de fluxo, informações estas que serão gravadas e exibidas. O sistema SCADA mantém um histórico sobre o desempenho total do ciclo que está a decorrer enquanto a malha de controlo, controla o ciclo que passa do PLC para o RTU.

A aquisição de dados neste tipo de sistemas inicia-se a nível do PLC ou do RTU e incluem o estado do equipamento e as leituras medidas, que são comunicadas como um requisito para o sistema SCADA. Os dados são aí formatados e compilados para que com o uso do HMI o operador, que se encontra na sala de controlo, possa fazer as decisões de supervisão, como reescrever ou ajustar o controlo normal dos PLC ou RTU.

Estes sistemas implementam maioritariamente uma base de dados distribuída, conhecida como uma base de dados do tipo *tag*, que contém elementos chamados de *points* ou *tags*. Um *point* é um único valor de *input* ou de *output* que está a ser controlado ou monitorizado pelo sistema e pode ser do tipo *hard* ou *soft*. O *input* ou *output* actual do sistema é representado por um *hard point*, enquanto que o *soft point* advém das operações matemáticas e lógicas que são aplicadas aos restantes pontos. Estes *points* são normalmente armazenados em pares: um valor e a data a que o ponto foi calculado ou gravado, que assim permite aceder ao histórico de um *point* em especial. O armazenamento de outros dados adicionais com as *tags* é também comum, como por exemplo comentários sobre o tempo de um projecto, informação sobre o alarme, caminho para o dispositivo no terreno ou no registo PLC [SCA, 2011].

2.1.2 Algumas Tecnologias relacionadas com EMS e sistemas SCADA

Nesta subsecção irão ser apresentados alguns Sistemas de Gestão de Energia existentes.

2.1.2.1 Uma Implementação para Controlo e Monitorização de Sistemas EMS/SCADA

Esta implementação constitui um exemplo de controlo e monitorização de um sistema do tipo EMS / SCADA, baseado nos *standards* internacionais, IEC 61850 e 61970. A solução proposta é baseada na tecnologia de *Web Services* que possibilita as vantagens de uma ferramenta livre, tais como: integração de aplicações; dissociação das aplicações; reduções nos custos de actualizações; reutilização de infra-estruturas existentes e de fácil desenvolvimento. As bases de dados do sistema foram construídas tendo em conta *Common Information Model* (CIM) do *standard* IEC 61970 e as interfaces normalizadas pela *Generic Interface Definition* (GID) também do *standard* IEC 61970. Já o *standard* IEC 61850 é usado para o desenho do *Substation Automation System* (SAS), que define a comunicação entre dispositivos na subestação e os restantes requerimentos do sistema.

A escolha da troca de todas as informações foi baseada num modelo de dados normalizado - normas IEC, através do uso de *Web Services*, com o intuito de maximizar as operações de rede centralizadas, tal como está apresentado na Figura 2.1.

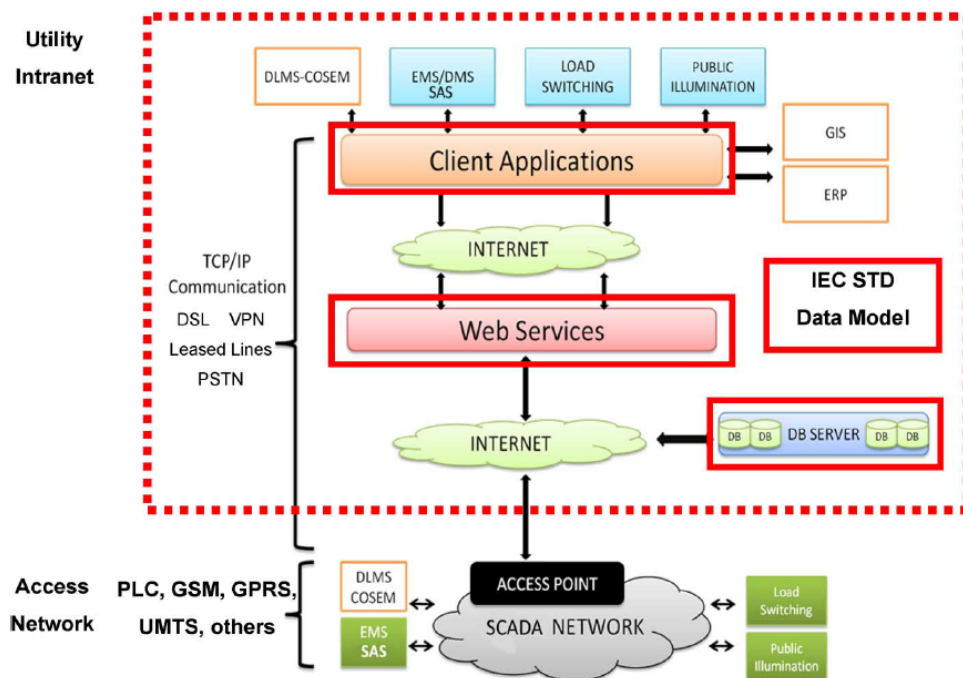


Figura 2.1: Implementação Proposta Baseada em *Web Services* e nos *Standards* IEC [Mercurio et al., 2009].

O principal objectivo do uso dos *standards* IEC já referidos é a minimização do esforço necessário para realizar a integração e para a realização de uma infra-estrutura de *software* flexível, isto é, o uso

destes *standards* garante a interoperabilidade entre aplicações e os seus modelos de dados e dentro dos protocolos de comunicação.

Pode-se concluir com esta implementação que os *standards* IEC referidos são fundamentais para o desenho, configuração e controlo de um *Energy Management Systems*.

Resumindo, pode-se dizer que os *Web Services* são mecanismos de integração onde os serviços dos *standards* são implementados. O uso dos *Web Services* como *middleware*³ permitiu obter muitas vantagens em termos de integração, reutilização de infra-estruturas existentes e de fácil desenvolvimento. O uso dos *standards* IEC melhora a interoperabilidade das aplicações, garante a flexibilidade e melhora o uso de infra-estruturas de integração [Mercurio et al., 2009].

2.1.2.2 O Futuro dos Sistemas SCADA para Sistemas de Distribuição Descentralizados

Os sistemas descentralizados de energia desempenham uma papel relevante no futuro dos sistemas de energia. Para tal, as infra-estruturas de electricidade do futuro devem suportar a integração do sistema de energia convencional com várias fontes de energia renováveis. Esta conexão entre as diversas fontes de energia tem ainda que permitir a monitorização e o controlo em tempo real.

A geração de energia e a não capacidade de armazenamento de electricidade é um problema que se vem discutindo tanto no passado como actualmente. Com o intuito de produzir energia limpa e de confiança, a integração das energias renováveis tem um papel importante para a resolução do problema. A tecnologia de geração de energia de pequena escala que fornece energia eléctrica ao consumidor e que está localizada no sistema de distribuição perto do local de consumo, chama-se *Distribution Generation* (DG). Como é mostrado na Figura 2.2, a energia gerada é totalmente consumida pelo produtor, ou então vendida à rede eléctrica mais próxima. O crescimento exponencial da geração de energia de pequena escala nas últimas décadas tem recebido maior atenção, não só a nível da sua vantagem económica como também a nível dos impactos ambientais que daí advêm.

Assim, o futuro dos sistemas SCADA deve garantir que se possam integrar os DG e as energias renováveis com a rede eléctrica comum. No entanto, a ampla dispersão geográfica onde estão localizadas as diferentes fontes de geração constitui um obstáculo que deve ser resolvido pelos DG de modo a garantir que os serviços disponibilizados são de confiança, tanto a nível da sua eficiência como no que diz respeito a garantir que a energia eléctrica tem o menor nível de poluição possível.

Os dois maiores benefícios da produção distribuída são os benefícios estruturais e benefícios relacionados com o mercado actual. Os benefícios estruturais lidam com a tentativa de minimizar os custos relacionados com a transmissão e com a energia através da combinação das diferentes fontes de energia. Os do mercado lidam com a volatilidade dos preços da energia, com a melhoria da qualidade da energia e com a fiabilidade em relação à mesma.

³É uma camada, usada em sistemas distribuídos, que é intermediária entre o *software* e as restantes aplicações.

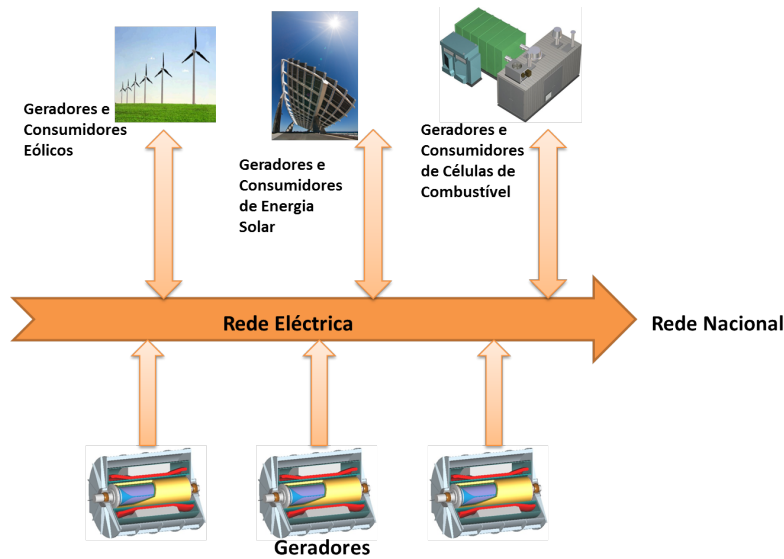


Figura 2.2: Diagrama de Blocos que representam um DG. Adaptado de [Geberslassie and Bitzer, 2010].

Os factores que levaram à produção descentralizada ou distribuída de energia assentam no facto deste método tornar o sistema inteiro de alimentação mais fiável, eficiente, flexível e económico [Geberslassie and Bitzer, 2010]. A Tabela 2.1 enumera as vantagens do uso da produção de energia distribuída em relação à centralizada.

Os sistemas SCADA do futuro terão assim que garantir a troca de informação e dados entre os diferentes nós na rede eléctrica inteira, que é constituída por *Energy Management Systems* e por *Distribution Management Systems*. Algumas das actividades realizadas pelos sistemas de gestão de energia consistem no controlo da transmissão, análise da rede, produção de energia e controlo; já o fluxo em tempo real, a verificação do estado da rede e o controlo do disjuntor são tarefas executadas pelo *Distribution Management Systems* (DMS).

Os sistemas DG usam um sistema de controlo descentralizado onde cada planta local de produção possui um sistema SCADA. Assim, no caso de alguma falha numa área de controlo, esta não irá afectar o sistema de energia inteiro. Os SCADA armazenam os dados medidos nos sistemas de energia através dos RTUs espalhados. Os *Master Terminal Unit* (MTU) e os IED actuam como sensores e estão localizados ao longo da rede com o intuito de guardar alguns dos dados medidos. Todos os centros de controlo comunicam depois com o sistema de controlo central.

O aumento da concorrência entre as empresas de energia eléctrica levará os sistemas de energia do futuro a crescer, tanto a nível técnico como económico. Logo, necessitará de maior capacidade computacional [Geberslassie and Bitzer, 2010]. A construção de infra-estruturas novas para substituir as antigas, deverá assim atender à necessidade dessa computação do futuro já que lidam com enormes quantidades de dados.

Tabela 2.1: Comparação entre os Sistemas Descentralizados e os Centralizados [Geberslassie and Bitzer, 2010].

PRODUÇÃO CENTRALIZADA	PRODUÇÃO DISTRIBUÍDA ou DESCENTRALIZADA
Perdas eléctricas elevadas para fornecer energia a utentes que estão distantes da produção da mesma	Reduz as perdas eléctricas, pois as energias renováveis estão perto dos clientes, logo tem menor distância de transmissão
O desenho das redes têm grandes limitações no que diz respeito a expansões futuras	É livre para futuras expansões
Devido à sua estrutura vertical (controlo central), em caso de falha ocorre uma falha de energia	Existe maior fiabilidade no fornecimento de energia, isto é, em caso de falha numa área de controlo esta é abrangida pelo centro de controlo central e o fornecimento de energia continua na rede eléctrica
Custos elevados das infraestruturas que permitam fornecer energia a locais remotos, tem ainda elevados custos de transmissão e perdas de energia	Mais económica em relação aos custos da rede de transmissão e nas perdas eléctricas
O fluxo de energia é unidireccional: da rede de produção para os utentes	Fluxo bidireccional: o utente pode produzir a sua própria energia numa pequena escala e também vendê-la à companhia eléctrica detentora da rede

No futuro, o sistema de energia terá um cenário diferente devido à grande quantidade de energia derivada do sector privado ou das plantas de geração de pequena escala. Este complexo cenário requer a troca de grandes quantidades de informação e grande poder computacional. Além disso, as redes de energia modernas possuem sistemas informatizados que fornecem informação eficiente para monitorizar e controlar a vasta gama de fontes de energia [Geberslassie and Bitzer, 2010].

2.1.2.3 Um Perfil para Electrodomésticos baseado nos EMS

Actualmente, a maioria dos sistemas de gestão de energia disponíveis no âmbito doméstico estão preocupados com a monitorização do consumo de energia em tempo real, com a visualização dos dados estatísticos e dos dados do consumo de energia em tempo real. Embora estes sistemas desempenhem um papel crucial no fornecimento de imagens sobre o consumo de energia num ambiente doméstico e influenciem nos comportamentos do consumo de energia, todos eles deixam os proprietários das casas como responsáveis por tomarem medidas adequadas no intuito de reduzir o seu consumo de energia. Estes EMS dão dicas sobre economia de energia, mas não têm em conta o perfil das famílias em questão. É neste ponto onde incide este projecto europeu de nome *Digital Environment Home Energy Management System* (DEHEMS), que pretende elucidar os proprietários das casas de habitação sobre como poupar energia, tendo também em conta os perfis de cada utilizador.

A arquitectura de alto nível representada na Figura 2.3, é baseada numa rede de sensores que medem o consumo de energia. Usa o protocolo *Zigbee*⁴ para troca de dados e criação das redes de sensores e com isto tem a vantagem de permitir que novos sensores que possuam o protocolo integrem a rede, caso seja necessário.

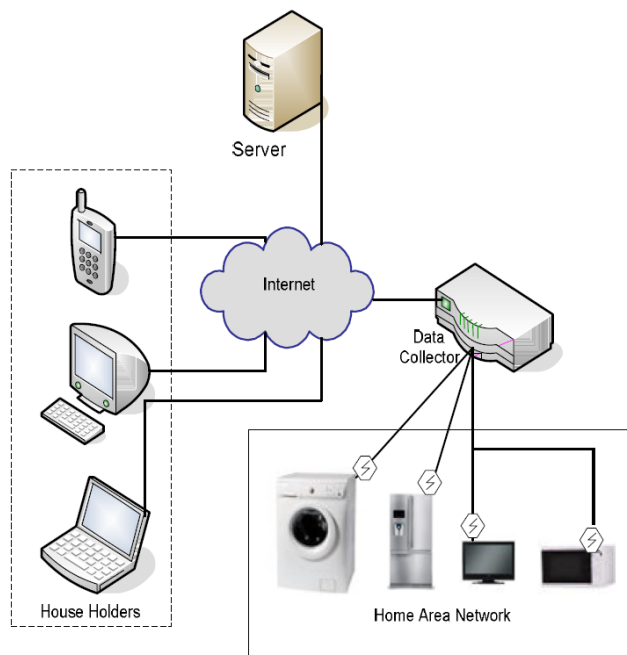


Figura 2.3: Arquitectura de alto-nível do DEHEMS [Chao et al., 2010].

A recolha de dados em tempo real no DEHEMS torna possível entender a correlação entre os electrodomésticos, análise estatística, geração de conselhos inteligentes e ainda diversos tipos de apoios a questões colocadas. Os proprietários podem também verificar o efeito das suas actividades de consumo em tempo real.

A arquitectura global do sistema proposto é mostrada na Figura 2.4 e cujas maiores componentes são um sistema baseado em regras e uma ontologia. O sistema fornece ainda uma interface com base nos *Web Services* para ser usado por outros sistemas que suportem o protocolo de troca de mensagens baseado em *Simple Object Access Protocol* (SOAP), ou por outras palavras, as funcionalidades do sistema estão expostas como serviços *web*. O *Protégé*⁵ é usado para a representação dos perfis dos electrodomésticos e para conselhos relativos aos consumos de energia baseados nas bases de conhecimentos e o *Jess*⁶ é utilizado pelo sistema como a entidade de raciocínio e basicamente, codifica o conhecimento empírico sobre as causas das anomalias e as recomendações úteis para contrapor essas anomalias.

Uma vez que, para diferentes tipos de utilizadores existem diferentes tipos de opiniões e preferências

⁴É usado para comunicações entre dispositivos electrónicos.

⁵É um editor de ontologias.

⁶É um ambiente para estabelecer as regras e os *scripts*.

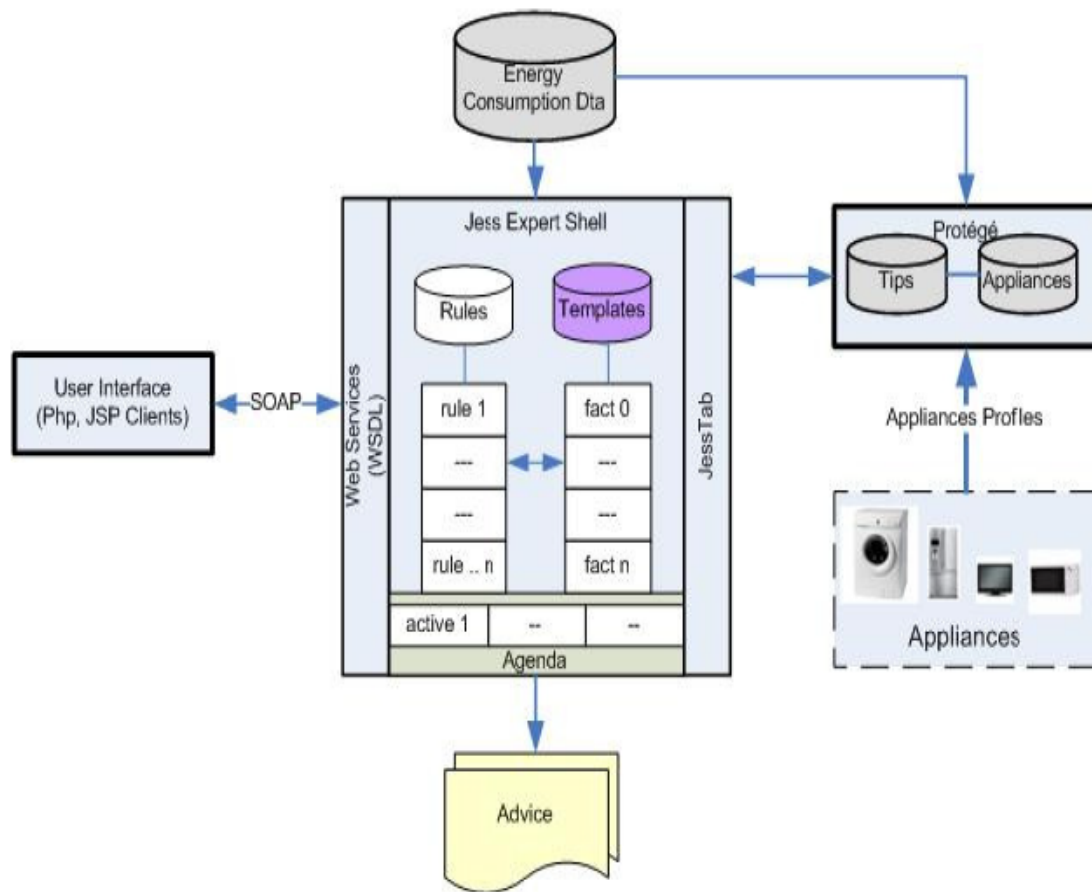


Figura 2.4: Arquitetura do Sistema DEHEMS [Chao et al., 2010].

e visto que o processo envolve vários critérios de avaliação, usou-se um método para reunir um consenso e para classificar as recomendações.

Em suma, este projecto consiste em utilizar os perfis de energia dos electrodomésticos para detectar e diagnosticar anomalias nos consumos de energia, recomendando depois soluções aos proprietários das casas com o sentido de remover ou minimizar essas anomalias. O sistema permite ainda aos seus utilizadores gravar toda a sua vivência sobre o sistema e o respectivo *feedback* [Chao et al., 2010].

2.1.2.4 Sistema de Supervisão e de Gestão de Energia para Grandes Edifícios Públicos

Os grandes edifícios públicos são importantes contribuintes para o consumo de energia de cerca de um terço da energia total consumida nas cidades, logo, o consumo de energia total pode ser reduzido com o uso dos *Building Energy Management Systems* (BEMS). Um sistema de supervisão e gestão para conservação de energia em grandes edifícios é apresentado em seguida, e funciona com um PC ou um sistema de computador embutido a executar um *software* BEMS desenhado para estar ligado a um sistema distribuído de medições de energia, a uma rede de sensores sem fios e até a BAS.

Os BEMS são sistemas que combinam inteligência e a tecnologia dos edifícios verdes, são edifícios

que usam um processo ambientalmente responsável em todo o seu ciclo de vida: localização, construção, manutenção, renovação e demolição. Para que sejam inteligentes, poupadores de energia e outros objectivos ambientais, a gestão unificada do consumo de energia é projectada para melhorar o funcionamento dos aparelhos e reduzir o consumo de energia. Os alvos desta operação são os sistemas de Aquecimento, Ventilação e Ar Condicionado, os sistemas de luzes, elevadores e equipamentos de escritório, etc.. Estes equipamentos, aquando da sua utilização, podem ser controlados por pessoas e pelo meio ambiente, para isso é necessário criar várias unidades de medidas e de controlo no edifício, pois quando o sistema se desvia temporariamente do funcionamento óptimo ou normal tem que ser corrigido de volta ao estado ideal. Posto isto, os seguintes resultados podem ser obtidos com o uso de um BEMS:

1. Melhoramento no nível de gestão do edifício.
2. Os equipamentos que não sejam eficientes podem ser encontrados.
3. Identificação dos consumos de energia anormais.
4. Redução dos picos de energia.

Pode-se então concluir que o principal objectivo dos sistemas BEMS é fazer com que o consumo de energia do edifício caminhe para um consumo perfeito, levando assim à conservação de energia e à alta qualidade do ambiente no interior do mesmo.

O *software* para gestão de energia dum BEMS necessita possuir uma boa colaboração multi-sistemas que seja capaz de fornecer uma variedade de dados em forma de objectos para possibilitar a integração e expansão do sistema dos edifícios. Já o *software* de supervisão dum BEMS deve fornecer tanto o modo C/S(Client/Server) para a gestão local, como B/S(Browser/Server) para gestão remota, que tradicionalmente são desenvolvidos separadamente, mas que podem ser desenvolvidos em conjunto usando as ferramentas indicadas, por exemplo o *Windows Presentation Foundation (WPF)*⁷.

Este projecto, apresenta assim, um novo sistema de supervisão e conservação de energia e também de estratégias para a gestão de energia em grandes edifícios públicos. Apresenta também melhorias na construção de um sistema de gestão de energia que foi projectado para manter um ambiente confortável e para poupar energia eléctrica. Os sensores sem fios foram implementados com o intuito de monitorizar ambiente interior e o BAS foi integrado no BEMS [Ma et al., 2010].

2.2 Síntese e Análise

Neste capítulo foram mostradas inúmeras situações onde a monitorização de dispositivos relacionados com energia é de verdadeira importância, tanto a nível de poupança de energia, como a nível da melhoria do aproveitamento da mesma e onde as *Information Technology (IT)* são usadas como ferramentas para o efeito. O NEMO-M apresenta-se assim como uma contribuição no suporte a um

⁷É a mais recente tecnologia da *Microsoft* para criação de interfaces gráficas.

sistema de gestão de energia que necessite de controlar dispositivos nas diversas áreas relacionadas com a energia.

Capítulo 3

Escalabilidade e Interoperabilidade nos Sistemas de Gestão de Energia

Este capítulo tem como objectivo mostrar a importância do uso de *Standards* no desenvolvimento de um projecto ICT, bem como enumerar os *Standards* que foram usados durante o desenvolvimento do projecto no qual a dissertação está inserida.

3.1 Interoperabilidade

A interoperabilidade é a capacidade de um sistema ou produto, de trabalhar com outros sistemas ou produtos, sem nenhum esforço em especial por parte do cliente. A interoperabilidade tornou-se uma característica com uma importância crescente para produtos ICT, por esta razão, o termo é amplamente utilizado na descrição de um produto [INT, 2011].

A inovação possibilitada pela interoperabilidade traz amplos benefícios para as sociedades que a venham a promover. Benefícios que advém da interoperabilidade [Gasser and Palfrey, 2007]: a interoperabilidade leva à inovação, que resulta em sistemas de tecnologia que trabalham em conjunto com mais facilidade, com menos problemas e garante que têm mais opções aquando da compra ou uso do mesmo. A inovação que advém deste tipo de interoperabilidade é vantajosa na medida em que aumenta o número de postos de trabalho e também os índices de produtividade das economias mundiais.

Segundo [Lee, 2010], dois sistemas de comunicação são ditos como interoperáveis quando trocam informação e usam essa informação como se ambos pudessem funcionar correctamente na mesma plataforma.

No processo de implementação das mensagens são adaptados *standards* relacionados para a interoperabilidade. Se não houver nenhum disponível, uma das partes do serviço poderá ajustar a sua implementação técnica à outra, ou então, ambas as partes podem acordar as suas especificações da implementação bilateralmente. No entanto, se houver um *standard* essas duas partes do serviço podem alcançar a interoperabilidade ajustando a sua implementação de acordo com o *standard*. A

qualidade da interoperabilidade inclui então uma boa adaptação ao *standard* e a conformidade com o *standard* existente.

3.1.1 *Service-Oriented Architecture*

SOA é um paradigma para organização e utilização de capacidades distribuídas que podem estar sob o controle de diferentes domínios.

Em geral, as entidades (pessoas e organizações) criam competências para resolver ou apoiar uma solução para os problemas que enfrentam no exercício da sua actividade. É normal pensar que as necessidades de uma pessoa possam ser solucionadas por recursos oferecidos por outra pessoa, ou no mundo da computação distribuída, os requisitos de um agente de computador possam ser atendidos por um agente noutro computador.

Não há necessariamente uma correlação um-para-um entre as necessidades e capacidades; a granularidade das necessidades e das capacidades variam do fundamental ao complexo, e qualquer necessidade pode exigir a combinação de diversas capacidades, enquanto qualquer capacidade única pode dirigir mais do que uma necessidade. O valor percebido do SOA é que ele fornece uma poderosa plataforma para adequar as necessidades com as capacidades e para a combinação das diferentes capacidades para lidar com essas necessidades [MacKenzie et al., 2006].

SOA é essencialmente uma colecção de serviços que podem comunicar entre si. Essas comunicações podem envolver uma simples transferência de dados ou uma coordenação de dois ou mais serviços [Barry, 2011].

SOA é um assunto vasto e cuja definição está longe de um consenso. No entanto, nas várias definições de SOA, encontram-se os seguintes tópicos [Ribeiro et al., 2008]:

- **Autonomia:** Não existe nenhuma dependência directa entre serviços.
- **Interoperabilidade:** É especificada a nível da interface, omitindo assim detalhes desnecessários.
- **Independência de Plataforma:** Os serviços são definidos usando o formato interoperável baseado em *eXtensible Markup Language* (XML).
- **Encapsulamento:** Os serviços fornecem propriedades próprias que são expostas por interfaces definidas pelo utilizador.
- **Disponibilidade/Descoberta:** Os serviços podem ser publicados em registos públicos e disponibilizados para uso geral.

Sendo o SOA um paradigma emergente para a modelação de sistemas distribuídos, é muitas vezes confundido com uma vasta gama de tecnologias de informação em rede. Neste contexto, os *Web*

Services são o mecanismo preferido para implementação de SOA [Ribeiro et al., 2008].

Uma das abordagens mais promissoras da arquitectura SOA é a aplicação a nível do dispositivo, onde o uso de infra-estruturas de comunicações baseadas em serviços de alto nível permitem avanços totalmente inovadores, melhorando significativamente a interoperabilidade e abertura desses dispositivos [Candido et al., 2009].

3.1.2 *Web Services*

Segundo [WS0, 2004], um *Web Service* é um sistema de *software* projectado para suportar interações interoperáveis entre máquinas numa rede. Tem descrita uma interface num formato que é processável por máquinas, o *Web Services Description Language* (WSDL). Outro modo de interagir com os *Web Services* é através de um modo estabelecido pela sua descrição usando mensagens SOAP, que são normalmente transmitidas através de *HyperText Transfer Protocol* (HTTP) com uma serialização XML, em conjunto com outros *standards* relacionados com *Web*.

A grande vantagem do uso de *Web Services* é que estes fornecem uma forma normalizada de interoperabilidade entre diferentes aplicações de *software* que podem estar a correr numa ampla variedade de plataformas. Como toda a comunicação é feita em XML, os *Web Services* não dependem de nenhum sistema operativo, nem de nenhuma linguagem de programação.

A arquitectura dos *Web Services* é uma arquitectura de interoperabilidade, isto é, identifica os elementos globais em toda a rede de *Web Services* que são necessários com o intuito de garantir a interoperabilidade entre os mesmos.

3.1.2.1 *Processo de Contratação de um Web Service*

Existem muitas maneiras para que a entidade requerente possa contratar e utilizar um *Web Service*, mas no geral, todas as etapas necessárias estão ilustradas na Figura 3.1 e são:

1. Ambas as entidades, requerente e servidora, tomam conhecimento uma da outra, ou pelo menos, uma conhece a outra;
2. As entidades referidas anteriormente concordam, de alguma forma, com a maneira como o serviço está descrito e sobre a semântica que irá reger a interacção entre o requerente e o servidor;
3. A descrição do serviço, bem como a sua semântica são realizadas pelos agentes requerentes e servidores;
4. Os agentes dos requerentes e dos servidores trocam mensagens, procedendo assim à realização de uma tarefa em nome das entidades requerentes e servidoras, isto é, a troca de mensagens com o agente servidor representa a manifestação concreta da interacção com o *Web Service* da entidade servidora.

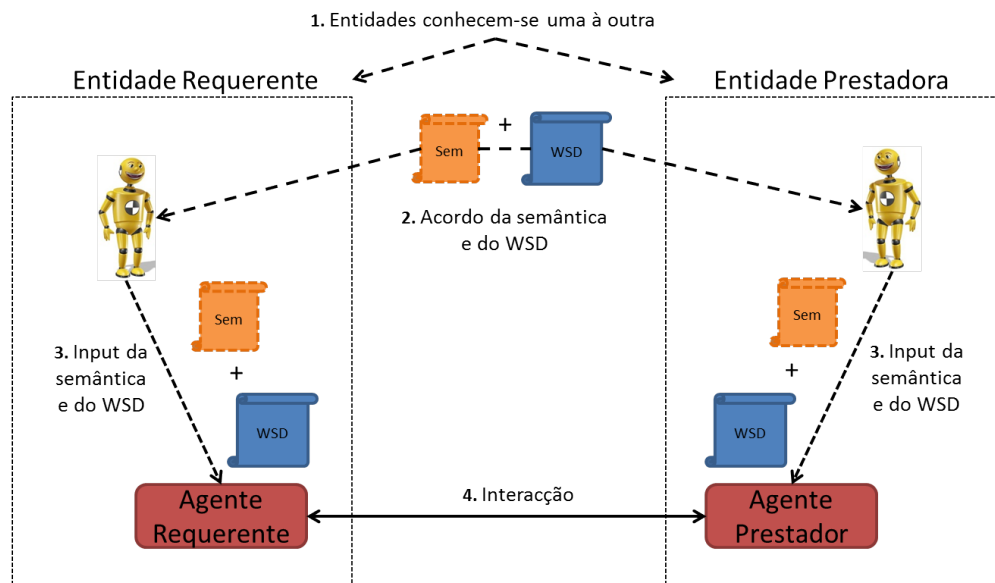


Figura 3.1: Processo Geral de Contratação de um *Web Service*. Adaptado de [WS0, 2004].

Faltando ainda referir que algumas destas etapas podem ser feitas automaticamente e outras podem ser executadas manualmente [WS0, 2004].

3.1.3 Importância do Uso de *Standards* na Indústria ICT

Na actualidade, existem cada vez mais expectativas por parte do público para a transparência, visualização e comunicação [Microsoft, 2009]. Para muitas empresas e organizações governamentais, ir de encontro a essas expectativas, significa explorar novas ferramentas e práticas que ajudam a fomentar a inovação, promover a eficiência e a criar benefícios económicos para os clientes finais. Os *standards* são um dos muitos instrumentos que podem ser utilizados para promover a interoperabilidade entre os produtos ou serviços dentro de um mercado e quando eles correspondem às necessidades reais, podem ajudar a promover a inovação, o crescimento do mercado e a proteger os investimentos em novas tecnologias.

Os *standards* podem catalisar a inovação, encorajar as empresas a contribuir com a sua tecnologia inovadora para as actividades para estabelecer novos *standards* e ainda partilhar a sua propriedade intelectual com os outros. Quando os *standards* são desenvolvidos e implementados por meio de processos abertos e transparentes, criam oportunidades para a diferenciação do produto e promovem mais escolhas para os utilizadores dos mesmos, aumentando desta forma a inovação e a competição entre os fornecedores.

Os *standards* têm um importante papel a desempenhar na indústria, pois o ecossistema ICT é acompanhado por uma contínua e rápida evolução dos seus constituintes e por uma forte competição entre empresas, que permite aos clientes uma escolha mais diversificada entre as numerosas soluções disponíveis.

As cinco principais características dos *standards* segundo [Microsoft, 2009] são as seguintes:

- Os *standards* são uma ferramenta para promover a eficiência, a interoperabilidade e a inovação.
- Os processos pelos quais os *standards* ICT são criados podem variar muito e estão constantemente em evolução.
- Pelo facto dos *standards* surgirem para cumprir as diferentes necessidades diárias dos consumidores, estão constantemente a ocorrer sobreposições dos *standards* actuais por outros mais recentes na indústria.
- As especificações da indústria não têm que se tornar *standards* formais para criarem o seu valor no mercado.
- Os *standards* por si só, não garantem a interoperabilidade, mas são uma mais-valia para atingir esse objectivo.

Como já foi referido, os *standards* são ferramentas que ajudam a promover a eficiência, a interoperabilidade e a inovação. De acordo com [Microsoft, 2009], os factores que contribuem para que o uso dos *standards* seja uma mais-valia para a indústria ICT são:

- **Eficiência:** Actualmente os utilizadores podem ver o seu trabalho feito de uma forma mais eficiente e eficaz devido aos produtos que utilizam frequentemente suportarem uma ampla gama de especificações e *standards*. No início da era dos computadores, muitos dispositivos, *software* e ficheiros de dados eram específicos para um fabricante de computadores, o que fazia com que o utilizador dependesse da comunicação entre máquinas do mesmo fabricante. A necessidade que um utilizador tem de comunicar ou usar dados de outras máquinas e/ou de outros utilizadores, evidencia que a mentalidade de silo existente entre os diferentes fabricantes de computadores torna esses procedimentos num processo demorado, pouco produtivo e frustrante para o utilizador. No entanto, hoje em dia já é possível fazer esses processos com maior facilidade, pois a maioria dos computadores e produtos de *software* já podem comunicar trocando dados directamente através de uma rede, usando um número ilimitado de computadores e dispositivos. O uso de *standards* contribuiu para que estes processos pudessem ser feitos.
- **Interoperabilidade:** Um meio para possibilitar os benefícios da eficiência e da acessibilidade é a interoperabilidade, permite comunicações muito diferentes e mais comuns entre dispositivos e serviços, permite também compreender os dados uns dos outros. No *software* este processo ocorre através do uso de protocolos para comunicação, ou via *Application Programming Interface* (API), através do qual o *software* recebe a entrada e fornece a saída. O *hardware* pode também incluir equivalentes físicos destas interfaces.
- **Inovação:** O aumento do desejo dos consumidores por novas experiências e mais funcionais, leva o mercado a novas invenções. Ao melhorar a capacidade do fornecedor para se concentrar na funcionalidade que define os seus produtos e os seus serviços para além do mercado, os *standards* ajudam a promover a inovação e o crescimento da tecnologia. Após as especificações ou os *standards* formais que criam uma plataforma comum sejam introduzidas no mercado,

os fornecedores concorrentes podem então focar-se no desenvolvimento de experiências realmente inovadoras para os utilizadores, tanto por introdução tecnológica de próxima geração, ou adicionando recursos complementares.

De modo a tornar o uso de *standards* o melhor suporte para melhorar a interoperabilidade, eles devem ser desenvolvidos e implementados através de uma abordagem multifacetada que inclui os seguintes passos:

- O processo de definição dos *standards* ser conhecido por todos.
- Manutenção pro-activa dos *standards*.
- Esforços industriais fortes e colaborativos, incluindo testes por parte dos fornecedores com o intuito de assegurar que diferentes implementações do mesmo *standard* sejam mesmo interoperáveis.

Ainda assim, a interoperabilidade nos ICT raramente significa uma implementação uniforme de um único *standard* [Microsoft, 2009].

3.1.4 *Standards* Usados no Trabalho Desenvolvido

De modo a cumprir os requisitos necessários para o bom funcionamento deste trabalho e pelos benefícios já referidos sobre o uso de *standards*, de seguida estarão descritos alguns dos *standards* que se optou por usar durante o desenvolvimento do projecto.

3.1.4.1 *International Electrotechnical Commission's 61850*

A integração de IEDs multi-funcionais numa subestação complexa ou na planta de um sistema de automação de energia, requer o desenvolvimento de um protocolo *standard* que vá de encontro com os requisitos de protecção, de controlo, de monitorização e de funções para gravar e medir os dados utilizados.

O *standard* emergente IEC 61850 para redes de comunicação e para sistemas em subestações permite o desenvolvimento de comunicações de alta velocidade baseadas nas comunicações *Peer-to-Peer* (P2P), bem como possui soluções para medições de dados distribuídos, soluções para controlo e protecção baseadas em amostras de valores analógicos [Apostolov and Vandiver, 2004].

Embora o IEC 61850 seja um *standard* de comunicação, o seu objectivo não é apenas realizar a troca de informações entre dispositivos, mas também o de tornar o sistema em questão interoperável.

De acordo com a definição do IEC 61850, a interoperabilidade é a capacidade para dois ou mais IEDs provenientes, ou não, do mesmo fornecedor, para trocar informações e usa-la para a execução correcta das suas funções específicas [Xu et al., 2007].

De seguida estão referidos alguns dos benefícios básicos, segundo [Hossenlopp, 2007], que advém do uso do *standard* IEC 61850:

- Vantagens em termos das velocidades de trocas de mensagens em relação a antigos *standards* usados para o mesmo efeito, permitindo assim que uma maior quantidade de dados sejam trocados no intuito de permitir um melhor funcionamento ou manutenção do sistema.
- Ligações P2P substituíram as ligações de fios convencionais sem *hardware* extra e ainda alargando a estrutura dos esquemas de automação de forma inovadora.
- As relações entre os clientes e o servidor passaram a ser mais flexíveis, de maior facilidade e com mais soluções para efectuar actualizações, comparando com as comunicações *Master-Slave* usadas anteriormente.
- Ficheiros de configuração XML que permitem trocas formais entre as diferentes ferramentas da engenharia, para desta forma, evitar dados de entrada duplicados, otimizar a consistência dos dados e reduzir os custos de engenharia.
- Certificado por diversos testes de conformidade de comunicação que ajudam a reduzir a variedade de interpretações encontradas nos diversos protocolos que levam a longos testes de integração e sintonia.

Em suma, o *standard* IEC 61850 oferece várias oportunidades para melhorar as operações e o controlo da rede, bem como o custo total de propriedade. Esta tecnologia já passou da fase inicial de implementação e como tal, já existe um número significativo de projectos a serem elaborados a nível mundial.

Para tirar total proveito das funcionalidades do *standard* em questão, as implementações dos projectos devem sofrer algumas alterações, ao invés de fazer uma "cópia" dos projectos com as tecnologias anteriores [Hossenlopp, 2007].

Nas comparações feitas por [Apostolov and Vandiver, 2004] entre o desempenho das soluções com fio e em termos de comunicações conclui-se que o IEC 61850 pode ser usado para substituir os sistemas convencionais de controlo e protecção das subestações sem qualquer degradação no desempenho global do sistema.

3.1.4.2 *Substation Configuration Description Language*

Uma secção do *standard* referido, a secção IEC 61850-6-1, especifica a SCL, que é baseada em XML e é usada para descrever a configuração dos sistemas baseados no IEC 61850. O SCL descreve uma hierarquia de ficheiros de configuração que possibilitam que sejam descritos vários níveis do sistema numa linguagem inequívoca e normalizada, como é o caso dos ficheiros XML.

Um cliente IEC 61850 pode extrair a configuração de um IED através do próprio quando estão ligados na mesma rede, no entanto, existem diversos cenários onde a disponibilidade de uma linguagem de

descrição formal *off-line* poderá trazer grandes benefícios aos utilizadores para além de configurar as aplicações do cliente IEC 81650. De entre os benefícios, destacam-se os seguintes:

- O SCL permite que as ferramentas de desenvolvimento do sistema administrem os ficheiros necessários para a configuração do IED em modo *off-line*, automaticamente a partir do sistema de energia, reduzindo assim os custos e o esforço na configuração do IED, através da eliminação da maioria, se não de todas, as configurações manuais.
- O SCL possibilita a partilha da configuração dos IEDs entre os utilizadores e os fornecedores para reduzir ou eliminar inconsistências e mal entendidos na configuração e requisitos do sistema. Os utilizadores podem ainda fornecer os seus próprios ficheiros SCL de modo a assegurar que os IEDs sejam entregues configurados correctamente.
- O SCL permite também que as aplicações IEC 61850 sejam configuradas em modo *off-line* sem a necessidade de uma ligação à rede com o IED para a configuração do cliente.

Este standard pode ser usado do modo desejado por cada utilizador, isto é, um utilizador pode, por exemplo, optar por usar ficheiros do tipo *Configured IED Description* (CID) de modo a ajudar na configuração do IED usando os processos existentes de configuração do sistema. O SCL pode também ser usado para reestruturar todo o processo de concepção do sistema de energia com o intuito de eliminar toda a configuração manual, eliminar os erros provenientes dos processos de entrada de dados manual, reduzir as possíveis falhas entre as capacidades do sistema e os seus requisitos, ajudar a melhorar a interoperabilidade do sistema final e ainda aumentar significativamente a produtividade e a eficácia dos engenheiros dos sistemas de energia [Baigent et al., 2003].

Embora os ficheiros SCL sejam ficheiros de texto e logo podem ser visualizados e alterados com qualquer editor de texto, esta abordagem requer um conhecimento bastante profundo da linguagem/standard XML e dos *schemas* SCL, existem ferramentas como por exemplo, o *SCL Visual Designer*, representado na Figura 3.2 que fornecem essas capacidades de visualização e alteração dos ficheiros SCL sem a necessidade do conhecimento da sintaxe XML subjacente [Huo et al., 2008], tornando-se por isso, uma mais valia na utilização deste tipo de ficheiros.

3.1.4.3 Abstract Communication Service Interface

Outra parte do *standard*, a secção IEC 61850-7-2, diz respeito ao ACSI que serve para descrever os serviços que o IED em questão possui e é especificado no *standard* em formato de texto e tabelas [Kostic and Frei, 2007].

Para permitir uma aplicação universal da infra-estrutura, usou-se um modelo de dados compatível com o *standard* IEC 61850 para que cada IED que estiver ligado à rede possa auto-descrever as suas características e funcionalidades através dum ficheiro SCL que detalha as suas ACSI. O uso de uma linguagem de comunicação comum a toda a infra-estrutura permite um reconhecimento e interacção dinâmicos entre cada dispositivo e o controlador da rede.

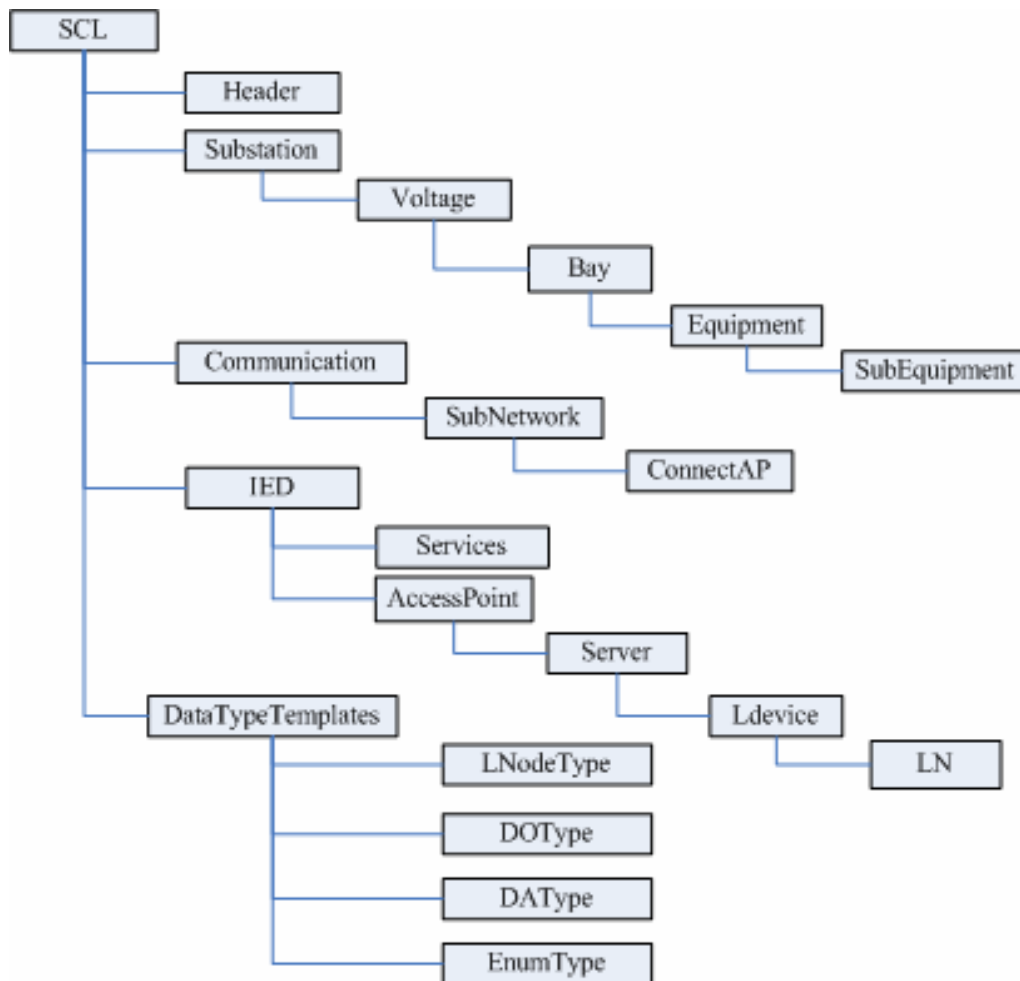


Figura 3.2: Estrutura Esquemática da Arquitetura SCL [Huo et al., 2008].

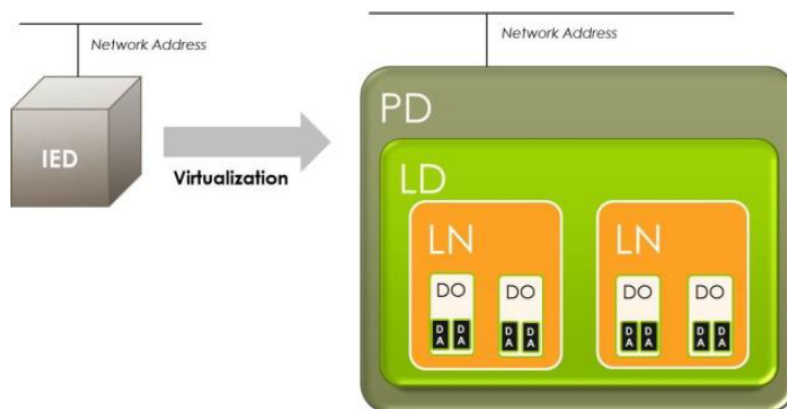


Figura 3.3: Virtualização de um IED usando ACSI [Lima et al., 2011b].

Como está mostrado na Figura 3.3, o IEC 61850 permite a virtualização de dispositivos reais, usando para esse efeito o modelo de informação do ACSI para descrever as funcionalidades específicas de cada IED. Cada um destes IEDs é decomposto em um ou mais *Logical Device* (LD), dependendo do tipo de dispositivos que o IED engloba, com cada LD herdando entidades de menor dimensão chama-

das *Logical Node* (LN), que representam diferentes tipos de características físicas. Cada LN contém vários *Data Object* (DO), que fornecem informações específicas sobre os respectivos LN. Para cada DO, existem vários *Data Attribute* (DA) que permitem o acesso às informações físicas específicas sobre o dispositivo. Para cada IED, os respectivos modelos de dados ACSI é descrito, usando para isso os ficheiros SCL definidos pelo IEC 61850 [Lima et al., 2011b].

3.1.4.4 *eXtensible Markup Language*

Actualmente o XML é a representação de troca de dados em rede mais importante e mais usada. O XML é um subconjunto do *Standard Generalized Markup Language* (SGML) que é uma forma estruturada para descrever os tipos de dados e permite ainda a criação de documentos para criar novas *tags*, no sentido de descrever com mais precisão todos os dados utilizados [Jing, 2010].

A crescente popularidade do XML é o resultado da sua flexibilidade e da sua força. Como já foi referido o XML permite criar as *tags* necessárias, o domínio da aplicação em causa torna o XML extensível, isto é, estende a capacidade de descrever um documento permitindo definir *tags* significativas para o caso de uso em questão.

Essa extensibilidade significa flexibilidade, mas a flexibilidade exige planeamento. Para fazer bom uso do XML é necessário conhecer e compreender todos os documentos a usar, quais as peças que os compõem, como elas se relacionam e como se pretende identificar as diferentes peças e é neste aspecto onde o modelo de objectos definido pelo *standard* IEC 61850 se torna a base para o desenvolvimento dos arquivos XML.

O XML é também uma linguagem *Markup*, porque a sua finalidade é identificar os elementos dentro de um documento. Sem marcação o computador vê qualquer documento como uma longa sequência de texto, com cada carácter a ter igual importância que outro carácter e que o facto de marcar o documento, permite que se identifique os diferentes elementos do documento numa maneira que lhe dá valor e contexto.

A grande vantagem do XML é a combinação destas duas características, a extensibilidade e as marcações, sendo por isso uma linguagem *extensible markup*, ou seja, pode-se marcar um documento de forma a satisfazer os requisitos da automação das subestações.

Considerando que a estrutura dos dados como objectos no IEC 61850 é hierárquica, torna-se então obvio que o XML está muito bem adaptado para o uso como um formato de arquivo SCL e onde essa hierarquia é definida pelas diferentes marcações.

Apesar das vantagens referidas sobre os ficheiros XML, um ficheiro bem definido não é suficiente, também é necessário haver algumas restrições que imponham regras que determinem a presença de elementos e os seus atributos, bem como a ordem desses elementos. Um exemplo dessas restrições passa pela existência dos XML *Schemas*. Uma propriedade importante deste tipo de ficheiros é o facto de também serem extensíveis, isto é, se necessário o *schema* pode ser alargado para aten-

der a novos requisitos. Esta linguagem é também conhecida como *XML Schema Definition* (XSD) [Apostolov, 2010].

3.2 Escalabilidade

Um sistema é considerado escalável se é possível ser implantado de forma económica e eficaz, abrangendo uma gama de diferentes tamanhos, devidamente definidos [Jogalekar and Woodside, 1998].

Muitos sistemas distribuídos devem ser escaláveis, o que significa que eles têm que ser economicamente implementados numa grande variedade de tamanhos e configurações. Aplicações típicas actuais e do futuro que incluem aplicações *web*, *e-commerce*, serviços multimédia, entre outros, devem poder ser utilizadas numa grande variedade de escalas em termos de número de utilizadores, número de serviços, quantidades de dados armazenados, cobertura geográfica, tamanhos de redes e de dispositivos de armazenamento. Escalas pequenas podem ser tão importantes como grandes escalas. A escalabilidade significa não apenas a capacidade de operar, mas operar de forma eficiente e com qualidade adequada ao serviço, sobre um dado intervalo de configurações. O aumento da capacidade deve então, ser proporcional ao custo e, a qualidade do serviço deve ser mantida [Jogalekar and Woodside, 2000].

3.3 Síntese e Análise

Como se pode verificar através dos inúmeros factos constatados ao longo deste capítulo, a adopção de *standards* para o desenvolvimento e elaboração do *NEMO-M* torna-o numa mais valia, tanto a nível de interoperabilidade como a nível de escalabilidade.

Capítulo 4

Solução Proposta

Este capítulo pretende apresentar ao leitor, uma descrição funcional e sucinta dos requisitos, da visão e dos constituintes do trabalho realizado.

4.1 Requisitos do Projecto

Nesta secção serão apresentados em primeiro lugar os requisitos do projecto NEMO e de seguida os requisitos do projecto NEMO-M que constitui esta dissertação.

O principal objectivo do projecto NEMO é desenvolver uma infra-estrutura de *software* baseada em *Web Services*, para ajudar a gerir e a controlar sistemas de energia bastante complexos. Tudo isto numa perspectiva de eficiência energética, onde sistemas de energia de fontes renováveis são usados na produção, distribuição e consumo de energia.

A infra-estrutura NEMO é suportada pelo paradigma SOA. Assim, um sistema compatível NEMO é uma colecção de sistemas, de subsistemas e de dispositivos encapsulados por serviços. Permitindo um acesso directo e integrado a informações da rede, com o nível de granularidade e semântica desejado, de modo a poder tomar as devidas decisões [Lima et al., 2010].

Para possibilitar a aplicação deste paradigma ao nível dos dispositivos existem inúmeros requisitos que se têm que cumprir, mas de modo a que estes permaneçam interoperáveis. A estratégia do NEMO para lidar com este requisito assenta em três principais pilares, nomeadamente SOA, *Web Services* e *Standards*, como está visível na Figura 4.1.

Assim, o paradigma SOA fornece a linha de fundo em termos de princípios e directrizes da arquitectura usada para apoiar a implementação da infra-estrutura do *software* NEMO. Os *Web Services* fornecem uma forma aberta e universalmente aceite para fornecer interoperabilidade entre sistemas heterogéneos. Por último, os *standards* possibilitam ao NEMO uma opção segura de evoluir e interagir com outros sistemas no futuro.

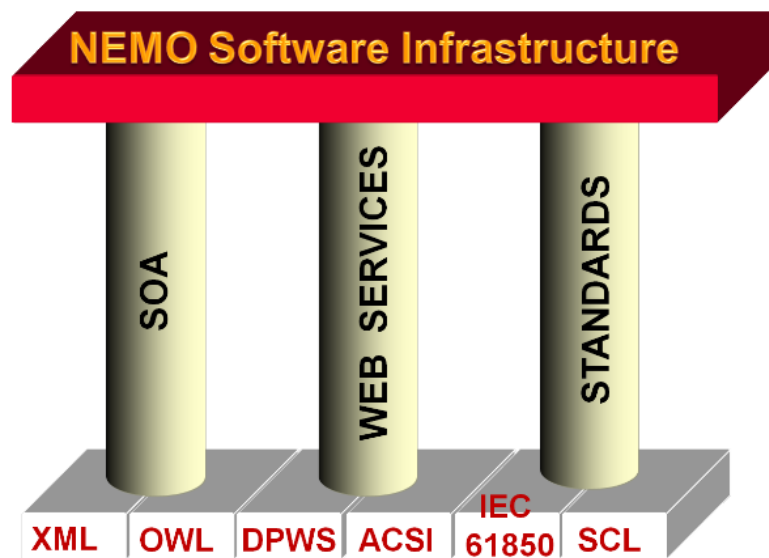


Figura 4.1: Os Pilares do Sistema NEMO.

Os requisitos do projecto NEMO podem ser divididos em duas categorias: os que são suportados por um sistema baseado no paradigma SOA e os genéricos. Os requisitos do projecto NEMO que obedecem ao paradigma SOA são:

- **Autonomia:** Nenhuma dependência directa entre serviços, estruturalmente dissociados.
- **Interoperabilidade:** Descrição dos serviços hospedados e dos padrões de interacção.
- **Independência da plataforma:** Baseado em *standards*.
- **Encapsulamento:** Funcionalidades embutidas expostas por interfaces, escondendo detalhes desnecessários.
- **Disponibilidade/Descoberta:** Serviços publicados nos registos públicos e disponibilizados para utilização geral.

Os requisitos que se podem denominar como genéricos, são:

- O sistema necessita ser *Plug and Play* (PnP) para permitir, tanto quanto possível, que um dispositivo fique disponível assim que entre na rede, isto é, para que a infra-estrutura suporte de forma transparente a inclusão de novos dispositivos permitindo uma gestão escalável do mesmo.
- Integração de sistemas e dispositivos desconhecidos através de protocolos de comunicação baseados em *standards*.
- Composição/orquestração dinâmica de serviços.

Como requisitos específicos do trabalho descrito por esta dissertação, tem-se:

- O trabalho terá que implementar e administrar uma infra-estrutura que possibilite a gestão de diversos componentes do núcleo do sistema. Estas componentes serão descritas com detalhe na secção 4.3.
- O trabalho tem ainda como requisito administrar todas as bases de dados do sistema. As bases de dados em questão serão explicadas com maior profundidade na secção 4.4.
- Para possibilitar a interacção entre o utilizador e o sistema, o trabalho tem também como requisito a elaboração duma interface para o efeito.

4.2 Solução Proposta: Visão Funcional, Visão Estrutural

A visão conceptual que guia o NEMO-M consiste em assumir que redes de dispositivos relacionadas com energia podem ser operadas, para isso serão ajudadas por uma infra-estrutura de *software* distribuído, baseado no paradigma de orientar os serviços e nos *standards*.

A Figura 4.2 representa de um ponto de vista mais específico ambas as redes que constituem o projecto NEMO, que já foram referidas na secção 1.4. O NEMO-M incidiu apenas sobre a rede de *software*, de onde se destacam os blocos representados na 4.2 como *NEMO Portal*, alguns dos *NEMO External Services* existentes, alguns dos módulos constituintes do *NEMO Kernel*, explicados em detalhe na secção 4.3 e ainda as três bases de dados representadas.

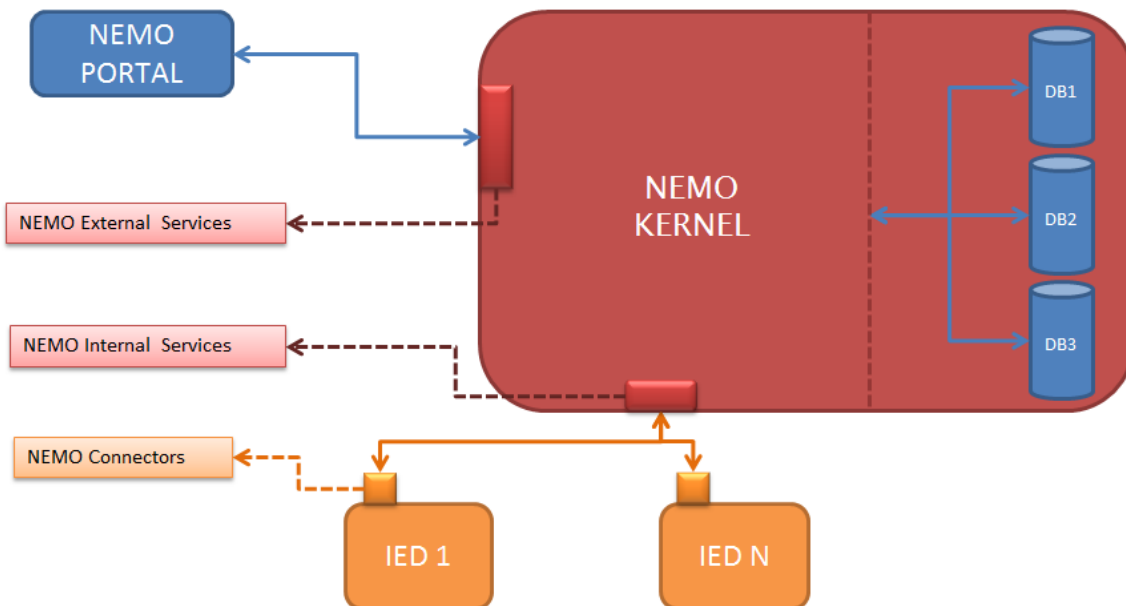


Figura 4.2: Vista do Sistema NEMO

4.2.1 Visão Funcional

A Figura 4.3, representa os casos de uso de um utilizador comum do trabalho desenvolvido, apresentando as funcionalidades que este dispõe.

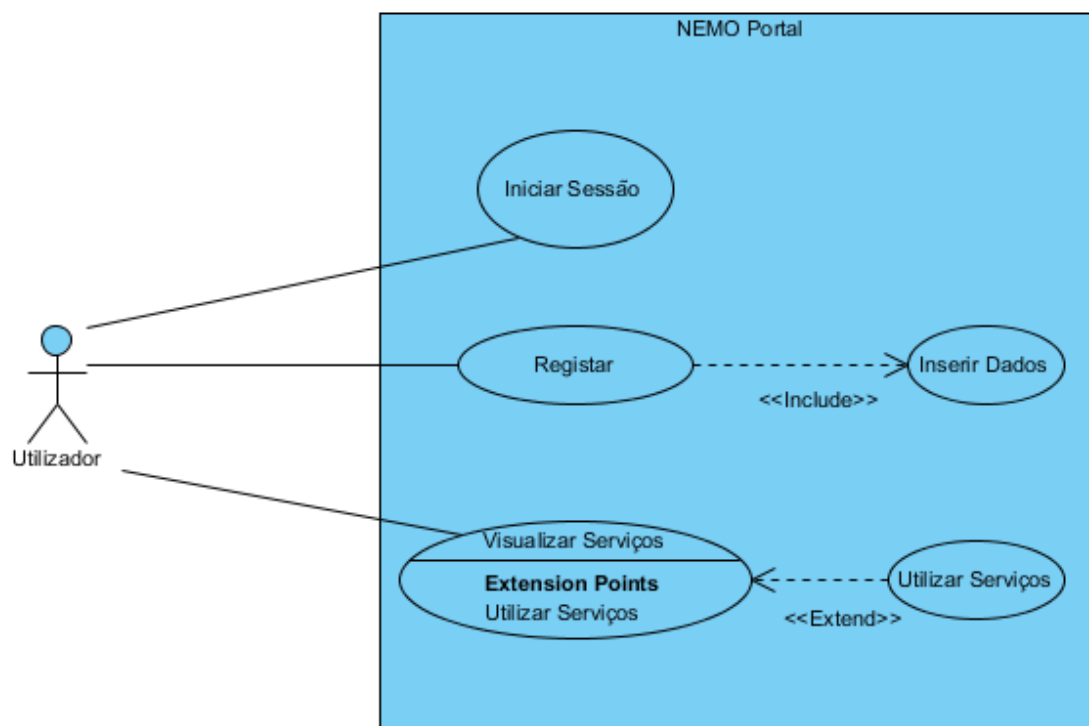


Figura 4.3: Diagrama Casos de Uso de um Utilizador

- **Iniciar Sessão:** permite a qualquer utilizador ter acesso às outras opções do sistema. Este *Login*, consiste na inserção no sistema por parte do utilizador, do *Username* e da *Password* correspondente, que posteriormente serão avaliados pelo sistema. Para que o *Login* seja possível, o utilizador terá que estar previamente registado no sistema.
- **Registrar:** possibilita a qualquer utilizador ter acesso à opção para pedir o registo no sistema. Após reencaminhado, o utilizador terá que inserir os seus dados pessoais: *Username*, *Password*, *Email* válido, entre outros, de modo a que o administrador do sistema possa avaliar o pedido. Caso aceite, o utilizador poderá então aceder ao sistema, através da opção *Login*.
- **Visualizar Serviços:** esta opção permite ao utilizador, após iniciar sessão no sistema, visualizar todos os serviços disponíveis para ele no mesmo. Posteriormente, o Utilizador poderá então usufruir de todos esses serviços.

A Figura 4.4, exibe os casos de uso do administrador da plataforma elaborada. Além das funcionalidades normais de um administrador, o diagrama mostra ainda as funcionalidades deste sistema em particular. Estas funcionalidades serão descritas detalhadamente em seguida.

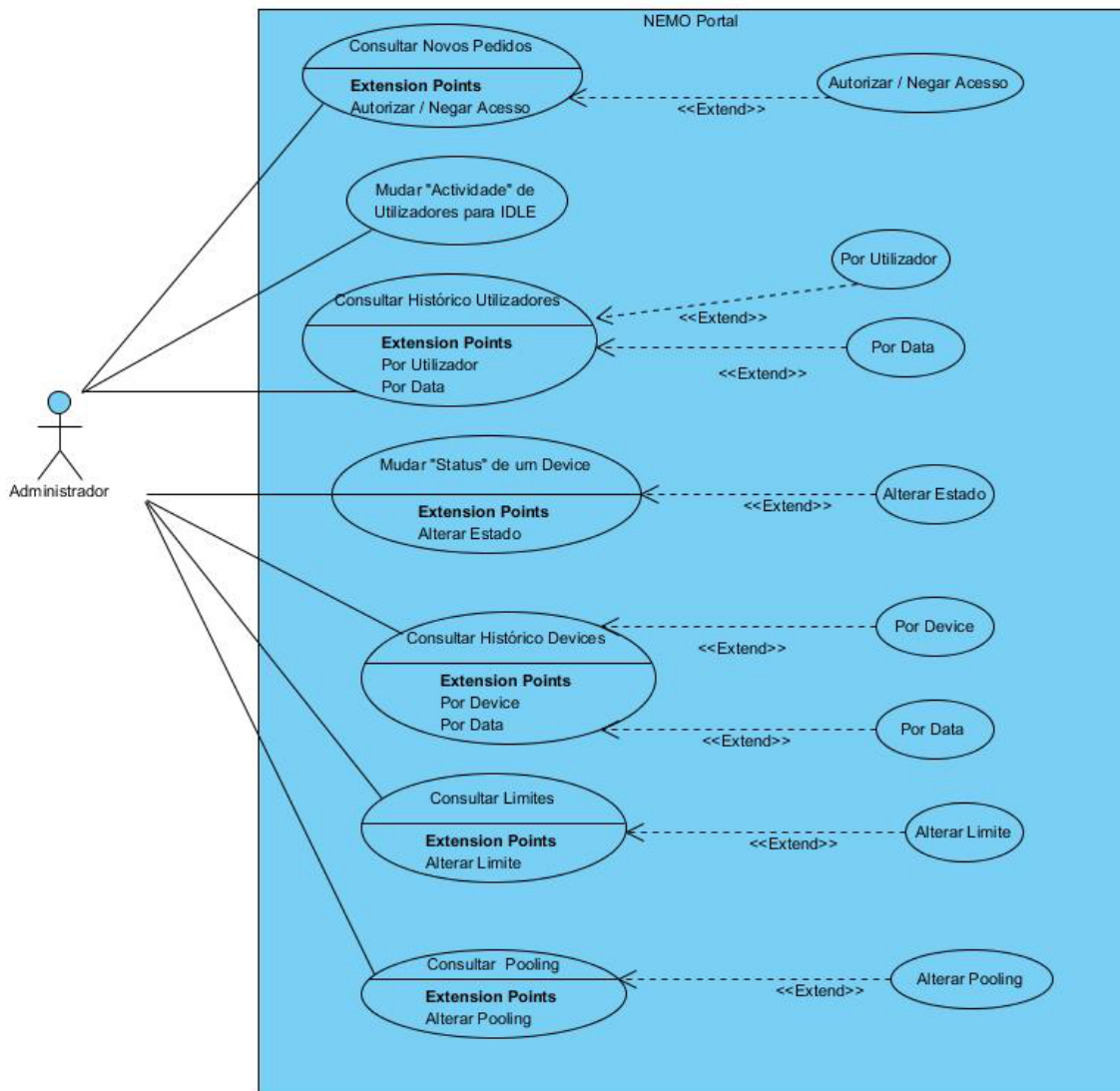


Figura 4.4: Diagrama Casos de Uso do Administrador

- **Consultar Pedidos Novos Utilizadores:** permite que o administrador visualize todos os pedidos, de possíveis utilizadores, para aceder ao sistema. Estes pedidos serão avaliados caso a caso e, após avaliação, o administrador tem a opção de enviar a cada um dos candidatos, a resposta ao requerimento para acesso, ou seja, se o utilizador foi ou não aceite no sistema.
- **Mudar Actividade de Utilizadores Para "IDLE":** possibilita ao administrador colocar como ausentes do sistema, "IDLE", todos os utilizadores que não tenham acedido ao mesmo no último mês. Este processo será executado automaticamente após ter sido escolhida esta opção. Esta será a única funcionalidade desta escolha, pois para o utilizador voltar ao estado normal, "ACTIVE", terá que voltar a aceder ao sistema.
- **Consultar Histórico Utilizadores:** esta funcionalidade permite visualizar o histórico de um utilizador do sistema em particular. Esta opção tem dois modos de pesquisa, permite visualizar o histórico total de um utilizador, e permite ainda observar o histórico de um utilizador restringindo a procura do mesmo a uma determinada data, ou mesmo a um determinado período.

- **Mudar "Status" de um Device:** esta funcionalidade permite ao administrador visualizar todos os dispositivos presentes no sistema, possibilitando posteriormente alterar o estado de um determinado dispositivo, mudando-o entre "ON" e "OFF", permitindo assim controlar os dispositivos que se pretende disponíveis para funcionamento na altura.
- **Consultar Histórico Devices:** esta característica do sistema possibilita ao administrador visualizar o histórico de um determinado dispositivo existente no sistema. A pesquisa dos dispositivos pode ser feita de dois modos: por dispositivo, que permite visualizar todo o seu histórico, ou restringida por uma determinada data, podendo ainda ser feita por um dado período.
- **Consultar Limites:** aqui o administrador pode observar os limites de produção, consumo e também do sinal de alarme ou emergência. Após consultados os valores, dispõe da opção de alterar qualquer um destes três valores.
- **Consultar Pooling:** o administrador pode ver os tempos de amostragem de alguns dispositivos e também o tempo com que alguns destes dispositivos gravam os seus dados em ficheiros, permitindo-lhe posteriormente alterar esses tempos.

4.2.2 Visão Arquitectural

A implementação do trabalho desenvolvido pode ser dividida em três camadas. Essa divisão de camadas corresponde à metodologia conhecida como Interface, Controlo e Entidade (ICE). O objectivo subentendido a esta metodologia é ter bem definidas e organizadas, a camada da implementação que diz respeito à *Interface* do sistema, ou seja, ao *Input / Output*, a camada que trata do controlo e processamento do sistema e a camada que cuida do armazenamento dos diversos dados usados.

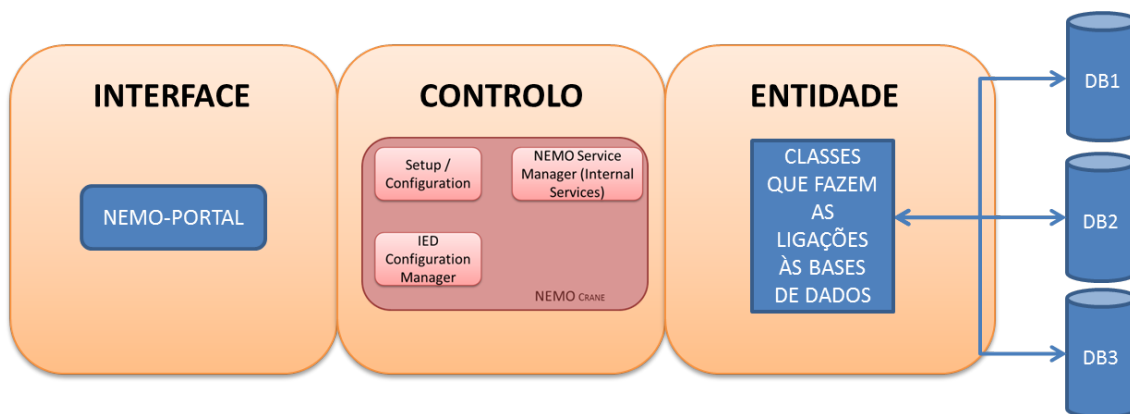


Figura 4.5: Relações ICE do NEMO-M

A Figura 4.5 ilustra de uma forma mais perceptível cada uma das camadas desta metodologia relacionadas com o trabalho desenvolvido, o NEMO-M.

A *Camada de Interface* que corresponde ao *NEMO Portal* (NEMO-P), é a camada mediadora entre o NEMO-M e os utilizadores. É esta que recolhe os pedidos do utilizador, *Input* do sistema, que posteriormente são conduzidos à camada encarregue pelo seu processamento. O resultado deste

processamento voltará novamente para a Camada de Interface, como *Output* do sistema.

A *Camada de Controlo* é responsável pelo processamento de toda a informação, pela avaliação dos dados, bem como pela escolha dos resultados a mostrar ao utilizador. Pode dizer-se então, que este é o principal bloco dos três existentes, pois comanda todas as operações, ou seja, gere todas as informações que passam pelo sistema. Este bloco, neste projecto, corresponde ao conjunto de componentes do *NEMO Kernel* (NEMO-K) representados na Figura 4.5.

A *Camada de Entidade* possui as classes que permitem a Camada de Controlo aceder às bases de dados do sistema desenvolvido. Permitindo consultar e alterar todos os dados guardados no sistema.

A Tabela 4.1 mostra alguns exemplos de relacionamentos que ocorrem no NEMO-M, entre as diferentes camadas da metodologia ICE num ponto de vista funcional, ilustrando dois exemplos que ocorrem no sistema. Estes exemplos são, neste caso, numa perspectiva de um utilizador comum.

Tabela 4.1: Exemplos de Relações entre as três Camadas - Utilizador

CAMADAS		
INTERFACE	CONTROLO	ENTIDADE
Funcionalidade Login	Verificação de dados	Classe " <i>Utils.java</i> "
	Veracidade dos dados	
Funcionalidade Register	Verificação de dados	Classe " <i>Utils.java</i> "
	Verificar Username	

Como pode verificar-se na Tabela 4.1, a *Camada de Interface* fornece ao utilizador uma funcionalidade, que se encontra na página web "*login.jsp*", que lhe permite efectuar o *LOGIN*, ou seja, esta interface possibilita que um utilizador solicite uma sessão no sistema NEMO-M. Este requerimento será efectuado através da inserção por parte do utilizador do *Username* e da *Password*. A *Camada de Controlo* tem como função, após recepção dos valores que são enviados pela *Camada de Interface*, validar todos os valores inseridos e, em seguida, verificar a sua veracidade, métodos estes que se encontram na classe "*LoginAction.java*". Esta fase do processo é efectuada em conjunto com a *Camada de Entidade*, isto é, a fidelidade destes dados só pode ser julgada após a procura dos mesmos na base de dados. Neste caso, é usada a classe disponibilizada pela *Camada de Entidade* para aceder à base de dados *Users*, a classe "*Utils.java*", verificando se o *Username* em questão existe e se a *Password* associada corresponde com o utilizador. A *Camada de Controlo* recebe então a resposta a essa avaliação proveniente da *Camada de Entidade*, e consoante a mesma, informa o utilizador através da *Camada de Interface* do resultado da tentativa de *LOGIN* no sistema.

A Tabela 4.1 ilustra ainda outra relação entre as diversas camadas. Neste caso a *Camada de Interface* possibilita a um utilizador, através da página web "*register.jsp*", que não esteja registado no sistema NEMO-M, fazer o registo para o mesmo. Este registo consiste na inserção, por parte do utilizador, de alguns dados pessoais na página referida, tais como: um *Username*, uma *Password*, um endereço de *email* válido entre outras coisas. Estes dados são transportados para a *Camada de Controlo*, que por sua vez valida todos os campos inseridos pelo utilizador, tendo ainda como função avaliar se o

4.2. SOLUÇÃO PROPOSTA: VISÃO FUNCIONAL, VISÃO ESTRUTURAL

Username pretendido pelo utilizador está ou não disponível para uso, estes processos encontram-se na classe *"RegisterAction.java"*. É neste processo, o de verificação da disponibilidade do nome de utilizador que entra a *Camada de Entidade*, que com o auxílio da classe *"Utils.java"* acede à base de dados dos *Users*. Por último, a *Camada de Controlo*, informa o utilizador através da *Camada de Interface* sobre a resposta do sistema. Ficando a resposta final/definitiva para o administrador do sistema.

A Tabela 4.2 representa alguns relacionamentos entre as diferentes camadas ICE, no ponto de vista do administrador do NEMO-M.

Tabela 4.2: Exemplos de Relações entre as três Camadas - Administrador

CAMADAS		
INTERFACE	CONTROLO	ENTIDADE
Funcionalidade ChangeActivity	Verificação de Utilizadores	Classe <i>"Utils.java"</i>
Funcionalidade UserHistory	Mostrar Todos Utilizadores	Classe <i>"Utils.java"</i>
	Verificação dos Dados	
	Nova Pesquisa	
Funcionalidade ChangeStatus	Mostrar Todos Dispositivos	Classe <i>"ServiceUtils.java"</i>
	Verificação dos Dados	
	Inserir na Base de Dados	

A *Camada de Interface* fornece ao administrador, através da página web *"success.jsp"* a possibilidade de mudar o campo *"Activity"* de um utilizador, isto é, a funcionalidade *ChangeActivity* que se encontra na Tabela 4.2. Esta informação após ser transmitida à *Camada de Controlo*, fará com que esta pesquise na base de dados dos *Users*, através da classe *"Utils.java"* disponibilizada para o efeito pela *Camada de Entidade*, todos os utilizadores da mesma. Em seguida, a *Camada de Controlo* irá avaliar um a um os utilizadores, para verificar qual, ou quais deles, se encontram inactivos no sistema há mais de um mês. Caso encontre algum, a *Camada de Controlo* irá novamente aceder à base de dados através da *Camada de Entidade*, para alterar o campo referido para *"IDLE"*. Por fim, e após verificação de todos os utilizadores, a *Camada de Controlo* informa o administrador através da *Camada de Interface*, na página web *"idle.jsp"* de todos os utilizadores que o campo foi alterado.

Outra das relações evidenciadas na Tabela 4.2, é a funcionalidade *UserHistory*, disponibilizada pela *Camada de Interface* na página web *"success.jsp"* e que permite ao administrador visualizar o histórico dos utilizadores. A classe *"SeeUserHistoryAction.java"* da *Camada de Controlo* ao receber este pedido, entra em contacto com a classe *"Utils.java"* da *Camada de Entidade*, que possibilita a interacção com a base de dados dos *Users*. Esta primeira interacção com a base de dados tem como objectivo obter todos os utilizadores registados no sistema. A informação é apresentada posteriormente ao administrador através da página web *"userHistory.jsp"* da *Camada de Interface*, para que o administrador fique com a possibilidade de escolher entre todos os utilizadores do NEMO-M, aquele que pretende ver o histórico. A escolha do administrador volta a ser transferida para a *Camada de Controlo*, que irá avaliar os dados na classe *"SeeUserHistoryAction.java"* e efectuar nova pesquisa nos utilizadores. Por último, a informação é delegada de novo para a *Camada de Interface*, desta vez a informação será visualizada pelo administrador na página web *"seeUserHistory.jsp"*.

Na última funcionalidade mostrada como exemplo na Tabela 4.2, a *Camada de Interface* oferece ao administrador, através da página web "*success.jsp*" a possibilidade de mudar o "*Status*" de um dispositivo, através da funcionalidade *ChangeStatus*. A *Camada de Controlo* recebe o pedido feito pelo administrador e a classe "*statusChangeAction.java*" entra em contacto com a *Camada de Interface*, mais propriamente a classe "*ServiceUtils.java*" para que esta aceda à base de dados dos *Services* e lhe disponibilize todos os IEDs que se encontram na rede do sistema NEMO-M, para que estes sejam mostrados ao administrador através da *Camada de Interface* na página web "*allDevices.jsp*". Com esta visualização, o administrador tem a possibilidade de escolher a qual, ou a quais dos dispositivos pretende alterar o "*Status*". Esta informação voltará a ser transmitida à *Camada de Controlo*, onde a classe "*statusChangeAction.java*" avaliará os dados recebidos e posteriormente voltará a entrar em contacto com a classe "*ServiceUtils.java*" da *Camada de Entidade*, para alterar o estado do dispositivo escolhido na base de dados. Por fim, a informação sobre as alterações efectuadas nos dispositivos do sistema serão novamente mostradas ao administrador, na mesma página web, ou seja, serão mostrados de novo todos os dispositivos com os estados alterados para clarificar o administrador sobre as mudanças efectuadas.

4.3 Constituintes do NEMO-K

Na Figura 4.6 encontra-se representado o NEMO-K, ou por outras palavras, a parte do sistema que vai gerir os diversos mecanismos existentes no sistema NEMO. Como se pode verificar o NEMO-K foi subdividido em dois conjuntos, o *NEMO Crane* e o *NEMO Brain*, sendo apenas o primeiro pertence ao sistema NEMO-M.

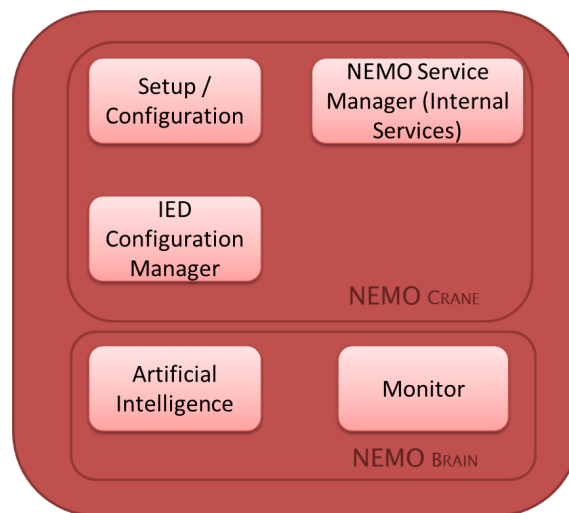


Figura 4.6: Vista Geral do NEMO Kernel

Os componentes do *NEMO Brain* como já foi referido não pertencem ao NEMO-M, fazem parte de trabalho futuro referente ao projecto NEMO como um todo. Onde componente *Artificial Intelligence* será o responsável pela inteligência artificial que o sistema irá conter e o componente *Monitor* será

responsável pela monitorização do mesmo.

Um dos requisitos do sistema NEMO-M consiste no desenvolvimento e administração dos componentes constituintes do *NEMO Crane*, que pertencem à *Camada de Controlo* do NEMO-M e passarão a ser descritos com detalhe em seguida.

4.3.1 NEMO Service Manager (Internal Services)

O componente *NEMO Service Manager (Internal Services)* é responsável pelas actualizações da base de dados dos *Services*, ou seja, este bloco terá que actualizar a base de dados sempre que um novo dispositivo entre na rede, acrescentando o dispositivo em questão e todos os serviços que dele advém, permitindo assim que os serviços disponíveis para o sistema no geral, sejam sempre actuais e de acordo com todos os dispositivos operacionais.

O modo de funcionamento deste componente é o seguinte: é feita uma pesquisa na base de dados dos *IEDs* procurando se houve entrada de novos dispositivos na rede, o mesmo quer dizer, que a base de dados terá novos serviços que necessitam ser avaliados e colocados *on-line*, para que o sistema possa dispor deles. Após esta pesquisa aos dispositivos, o procedimento irá procurar por todos os serviços que este IED dispõe, procurando pelos serviços *Standard Compliant* e pelos que não obedecem ao *Standard*, guardando-os numa estrutura auxiliar, colocando em seguida essas informações na base de dados dos *Services*, associando assim os serviços ao IED correspondente.

Em suma, *NEMO Service Manager (Internal Services)* limita-se a ir buscar informações sobre os serviços à base de dados dos *IEDs*, guardando-as em estruturas auxiliares, passando-a em seguida para a base de dados dos *Services*. Este componente apenas tem interacção com as duas bases de dados referidas.

4.3.2 IED Configuration Manager

IED Configuration Manager é o componente responsável pela administração da base de dados dos *IEDs*. Esta administração consiste na inserção de novos dispositivos sempre que apareça algum novo na rede do sistema.

Este módulo do *NEMO Crane* funciona recebendo como parâmetros de entrada dois ficheiros, o ficheiro SCL e o ficheiro *NEMO File* (NEMO-F). O ficheiro SCL, baseia-se no SCL, que é a linguagem especificada pelo IEC 61850 para a configuração de dispositivos eléctricos. Visto obedecer ao *Standard* e permitir transmitir grande parte das informações sobre o IED em questão, optou-se por usá-lo no sistema. Por outro lado, e para bom funcionamento do sistema, foi criado o NEMO-F, que contém as informações que o SCL não suporta, como por exemplo serviços que o dispositivo possua que não estejam contemplados no *Standard*. Permitindo assim guardar na base de dados dos *IEDs* todas as informações que o sistema NEMO necessita relativas ao dispositivo.

O componente é lançado por um dos serviços dos *NEMO Internal Services*, que, após receber os dois ficheiros já referidos, provenientes do IED, o lança. O *NEMO Connector* (NEMO-C) que está colocado no IED em questão, quando este entra na rede do sistema, identifica-se e envia os ficheiros através de *Devices Profile for Web Services* (DPWS), usando o serviço indicado para tal. Após recepção dos ficheiros, lança este bloco, que fica responsável pelo tratamento dos mesmos. Na parte de saída do módulo está a base de dados dos *IEDs*, ou seja, as informações contidas em ambos os ficheiros são tratadas e inseridas na base de dados.

4.3.3 Setup / Configuration

A funcionalidade deste bloco é a de instalar, configurar e inicializar todos os constituintes do NEMO-M.

4.4 Diagrama de Entidades e Relações (DER)

Os Diagrama de Entidades e Relações (DER) são diagramas que permitem representar o modelo de dados do sistema numa perspectiva de alto nível, possibilitando assim ao criador do sistema ter uma noção mais concreta da base de dados do sistema.

Na criação deste sistema em particular, o NEMO, foi necessário recorrer à criação de três bases de dados para o seu bom funcionamento. Assim, foram criadas as bases de dados dos *IEDs*, dos *Serviços* e dos *Users*, base da dados essas que terão o seu DER representado e explicado em seguida.

4.4.1 Base de Dados *IED*

A base de dados em questão foi criada com o objectivo de guardar todas as informações necessárias sobre cada IED existente no sistema. Visto cada um destes dispositivos ter as suas informações guardadas tanto no ficheiro *SCL* como no NEMO-F, este *DER* foi criado, tendo em vista possibilitar que todas as informações contidas em ambos os ficheiros pudessem ser armazenadas na base de dados.

Tendo em conta que este *DER* é relativamente complexo, e com o intuito de o tornar mais perceptível, decidiu-se representa-lo dividido em quatro figuras (Figuras 4.7, 4.8, 4.9, 4.10).

Na Figura 4.7 encontra-se representado o que se pode chamar de cabeçalho do ficheiro *SCL*. A entidade representada como "*SCL*" guarda as informações sobre o cabeçalho do ficheiro *SCL*, tem uma ligação de 1 - N com a entidade "*IED*" pois pode conter mais que um IED no mesmo ficheiro. Esta entidade "*IED*" faz a ligação entre as informações contidas no ficheiro *SCL* e as contidas no NEMO-F. A ligação N - 1 com a entidade "*ACCESSPOINT*" indica que o mesmo ponto de acesso para um dispositivo pode estar disponível em mais que um IED.

4.4. DIAGRAMA DE ENTIDADES E RELAÇÕES (DER)

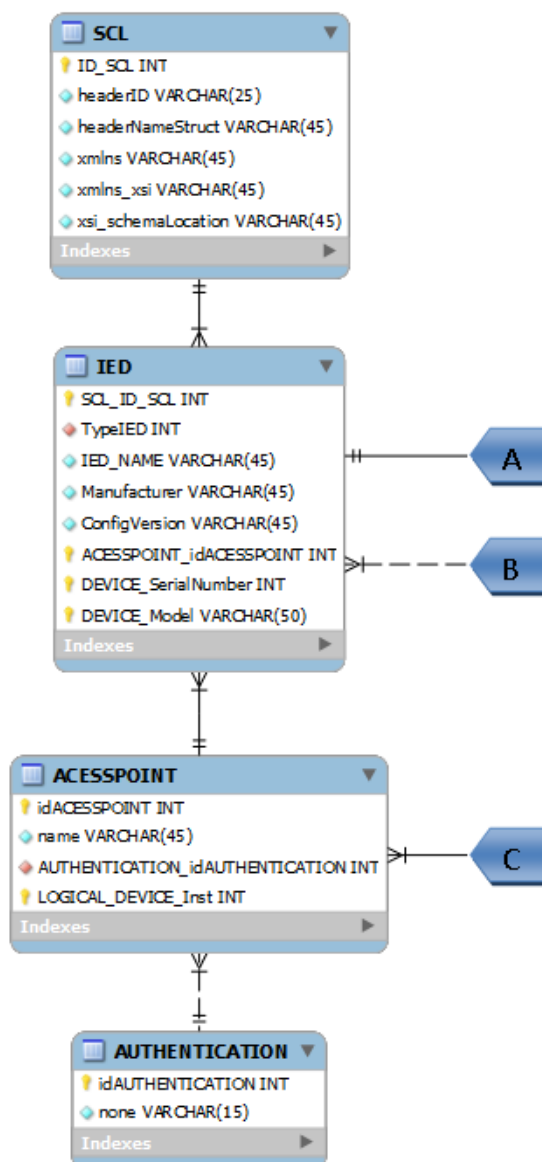


Figura 4.7: Diagrama de Entidade e Relação da base de dados IED - Componente SCL Heading

Falta ainda referir que as ligações que terminam com as etiquetas "A" e "B" representadas na 4.7 tem continuidade na 4.8. E a ligação da etiqueta "C" continua na 4.9.

A Figura 4.8 ilustra a componente do DER que corresponde à informação proveniente do NEMO-F, ou seja, as tabelas aqui representadas foram criadas para permitir o armazenamento de todas as informações que daí advém.

Uma das entidades representadas na Figura 4.8 é *NemoFile* que corresponde ao cabeçalho do NEMO-F, tem uma relação 1 - N com a entidade "DEVICE" que identifica o dispositivo e por sua vez uma relação N - M com a entidade "SERVICE". Esta relação indica que para o mesmo dispositivo podem existir mais que um serviço e ao mesmo tempo, o mesmo serviço pode existir em mais que um dispositivo.

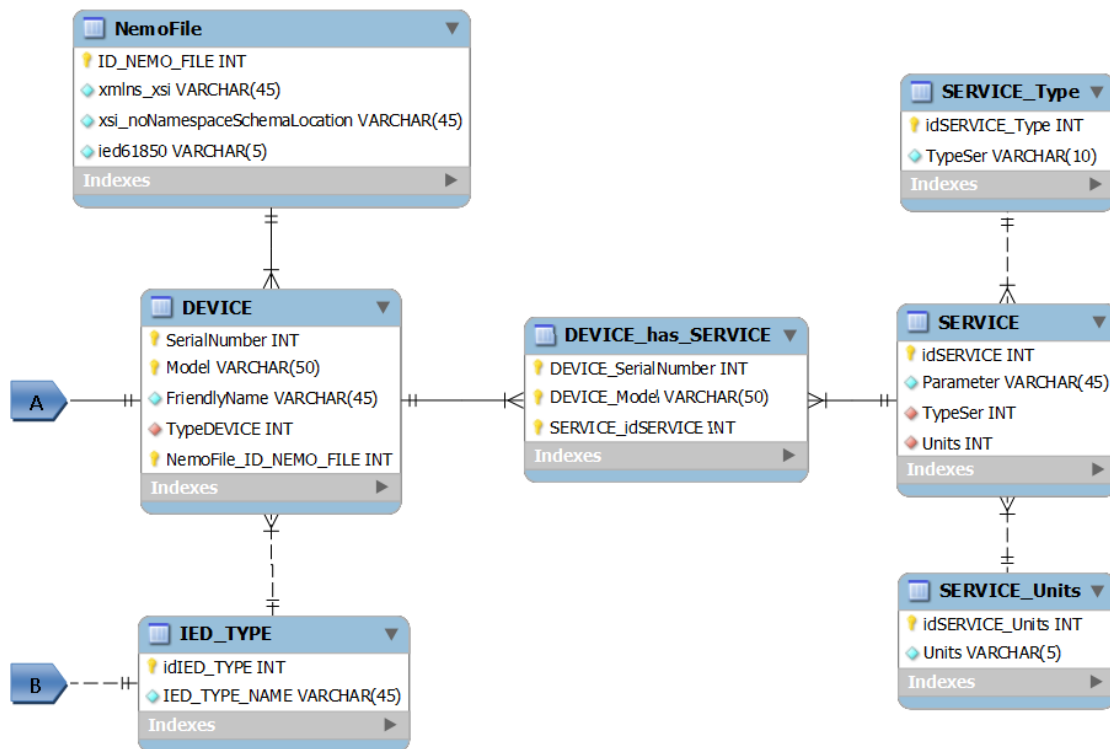


Figura 4.8: Diagrama de Entidade e Relação da base de dados *IED* - Componente NEMO File

Na Figura 4.9 está exibido a parte do *DER* que representa o dispositivo físico no ponto de vista do *IEC 61850*, possuindo todas as informações relativas ao IED que o formato SCL suporta.

A entidade *"LOGICAL DEVICE"* tem uma ligação N - M com a entidade *"LOGICAL NODE"*, que por sua vez possui o mesmo tipo de relação N - M com a entidade *"DOI"*, *Data Object Information* e este a relação N - M com a entidade *"DAI"*, *Data Attribute Information*.

De referir que a ligação que está descontinuada e representada com a etiqueta "D", terá a sua continuação na Figura 4.10, com a ligação à tabela correspondente.

A última figura que representa este *DER*, a Figura 4.10, contém os tipos de data/objectos que o *IED* em questão possui, assim como todas as informações relativas a cada um desses tipos.

4.4.2 Base de Datos *Services*

Esta base de dados foi criada com o intuito de relacionar os *IEDs* existentes no sistema com os Serviços que cada um deles disponibiliza para o mesmo. Assim, a Figura 4.11 ilustra o *DER* que representa a base de dados nomeada *Services*.

Na Figura 4.11 encontra-se então representada a ligação entre as entidades *DEVICE* e *SERVICE*, relação esta que é do tipo *N-M*, isto é, um *DEVICE* pode oferecer mais que um *SERVICE*, e um *SERVICE* pode ser disponibilizado por mais que um *DEVICE*.

4.4.3 Base de Dados *Users*

Esta base de dados serve para armazenar todos os utilizadores existentes no sistema, isto é, qualquer utilizador que pode, ou tentou, aceder ao sistema NEMO está registado nesta base de dados.

Na Figura 4.12 está representada a entidade *NemoUser* que armazena todas a informações sobre cada um dos utilizadores registados. De seguida passa-se a explicar as informações que o *DER* contém:

- *idUser* - Armazena o nome do utilizador, sendo a chave primária da entidade.
- *Password* - Guarda a *password* do utilizador.
- *Email* - Contém o endereço de *email* válido do utilizador.
- *PermissionLevel* - É responsável pelo nível de permissão que o utilizador tem no sistema.
- *Status* - Guarda o estado utilizador no sistema relativamente ao seu pedido de registo. Há três estados possíveis, *WAITING*, *REGISTERED* e *REJECTED*.
- *Position* - Posição que o utilizador do sistema aufere no mesmo, ou de *ADMINISTRATOR*, ou *OPERATOR*.
- *CompanyName* - Nome da empresa onde o utilizador está empregado.
- *LastAccess* - Conserva a data da última vez que o utilizador acedeu ao sistema.
- *Activity* - Arquiva o estado da actividade do utilizador em questão. Dois estados possíveis *ACTIVE*, ou *IDLE*.
- *HistoryFile* - Ficheiro que contém o histórico de todas as acções do utilizador.

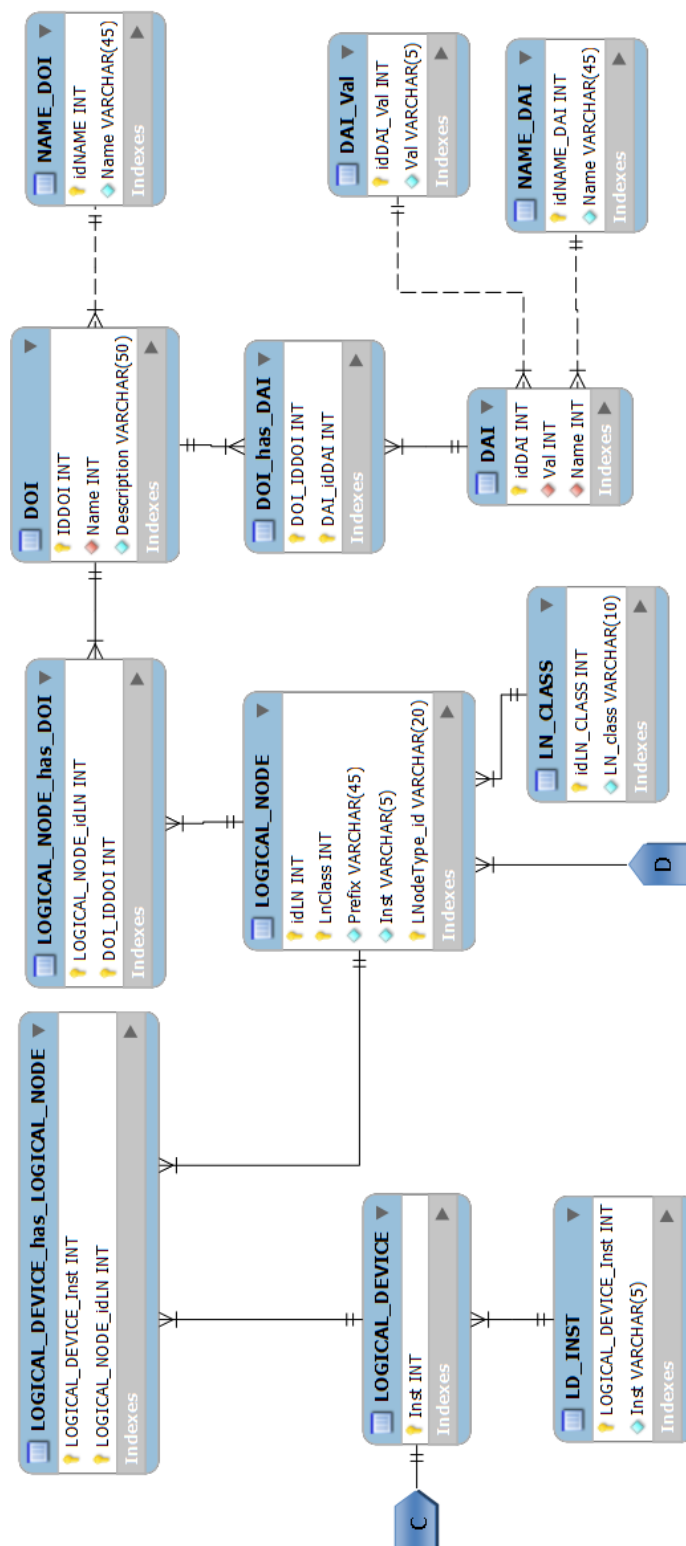


Figura 4.9: Diagrama de Entidade e Relação da base de dados IED - Componente SCL

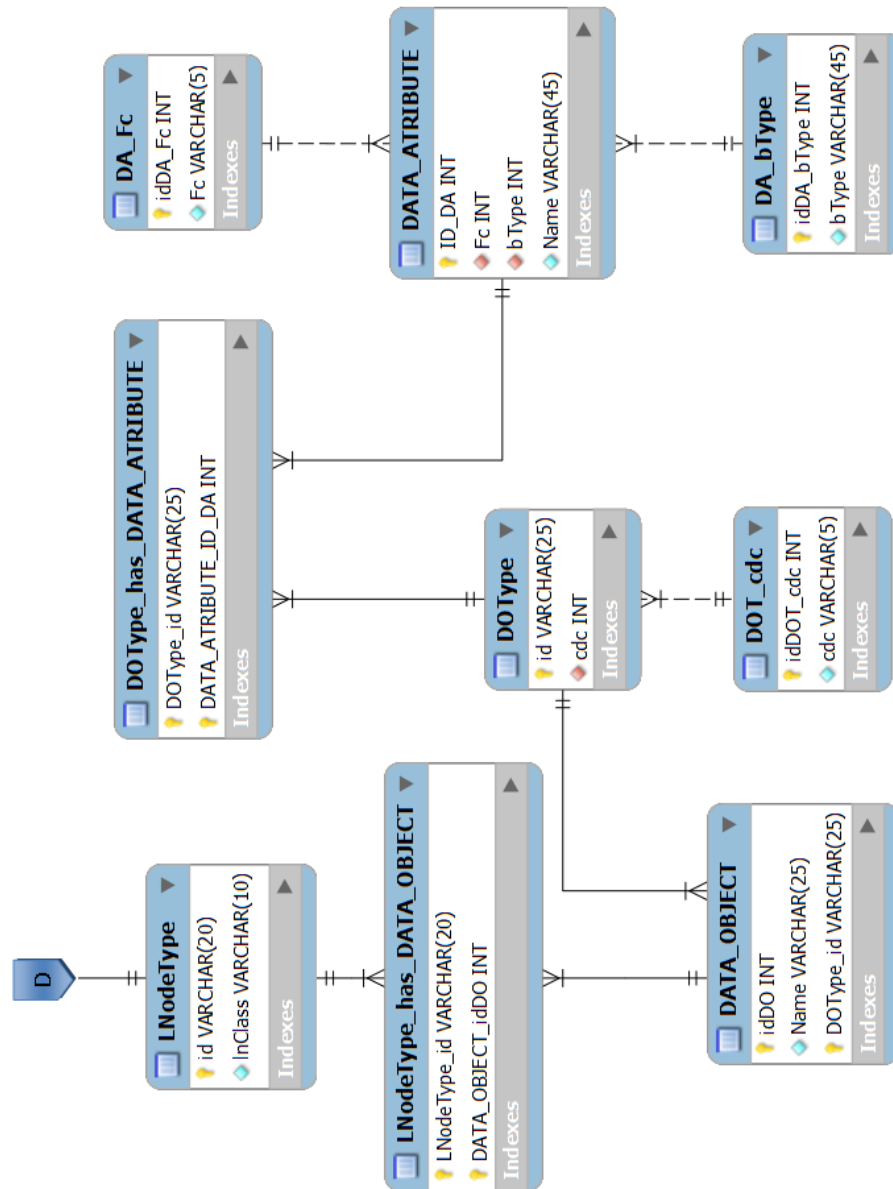


Figura 4.10: Diagrama de Entidade e Relação da base de dados IED - Componente *Data Type Templates*

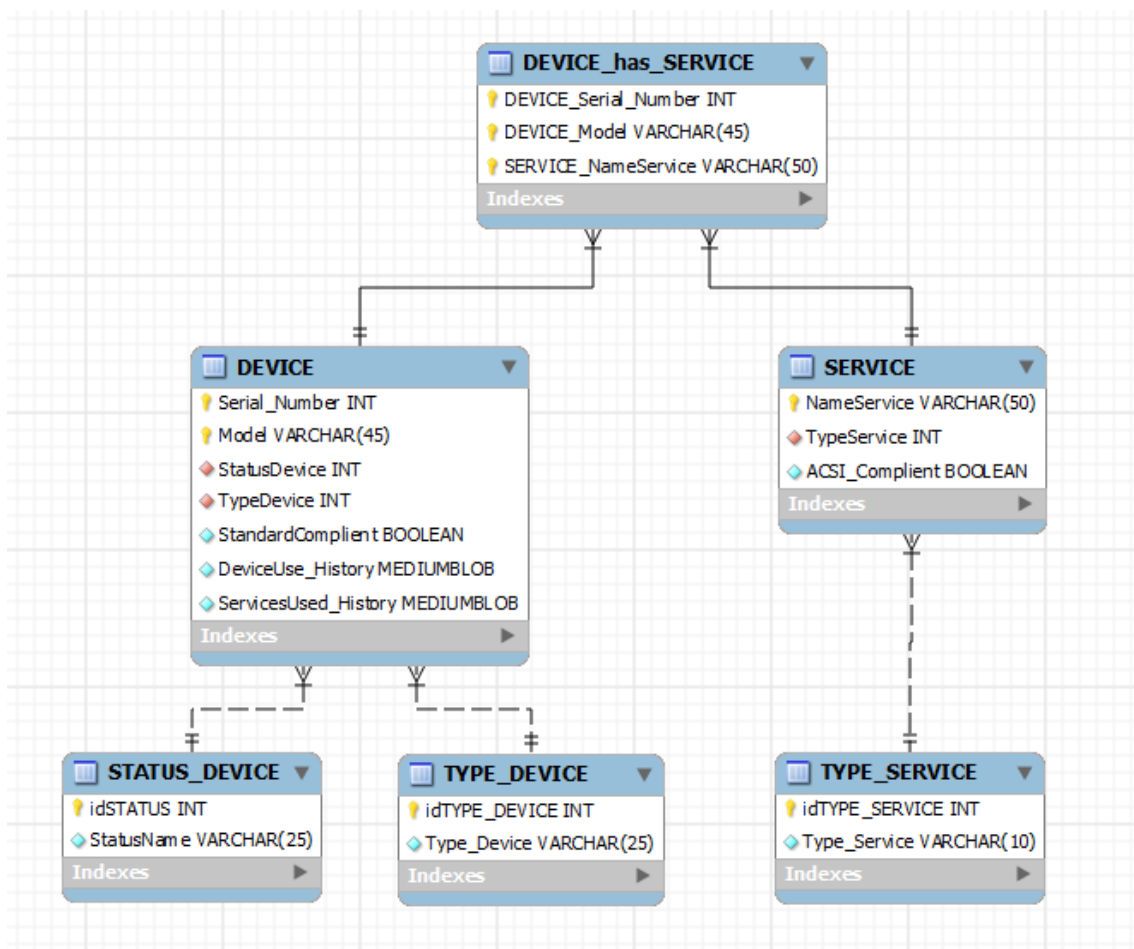


Figura 4.11: Diagrama de Entidade e Relação da base de dados *Services*

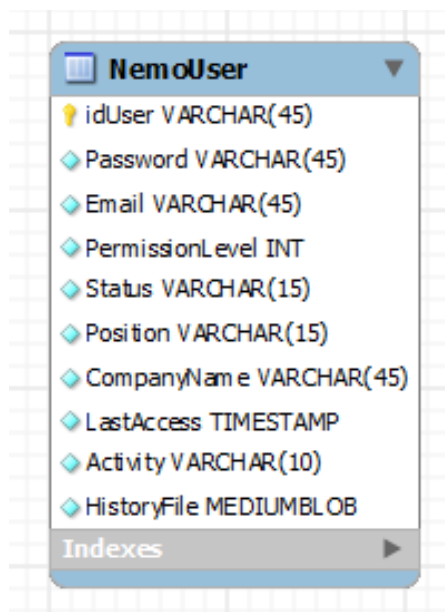


Figura 4.12: Diagrama de Entidade e Relação da base de dados *Users*

Capítulo 5

Implementação

Este capítulo retrata a implementação do sistema NEMO. Serão referidos aspectos relacionados com as tecnologias utilizadas, assim como a maneira como certas trocas e processamento de informação foram efectuadas no decorrer do projecto.

5.1 Escolhas Tecnológicas Feitas no Decorrer do Projecto

Nesta secção serão abordadas as escolhas tecnológicas efectuadas no decorrer do desenvolvimento da plataforma de *software* elaborada, bem como as vantagens no uso das mesmas.

Para a base do projecto foi usada a linguagem de programação *Java*, com o auxílio do *Integrated Development Environment* (IDE) *Netbeans*. O recurso ao *Java* foi feito devido à necessidade do uso do DPWS, que só estava disponível em duas linguagens, *C* e *Java*, sendo que o *JAVA* era a linguagem de mais alto nível, permitindo o recurso a classes. Assim, algumas das tecnologias adoptadas advém dessa linguagem de programação, como é o caso do *JavaServer Pages* (JSP), *Struts*, *Hibernate* e ainda a biblioteca *JDBC*. Tecnologias estas que serão descritas nesta secção.

A *Camada de Interface* do sistema foi elaborada em JSP. O JSP é uma tecnologia de servidores do *JAVA*, que permite a quem desenvolve o *software*, criar páginas *Web* dinamicamente geradas, com *HyperText Markup Language* (HTML), XML, ou outro tipo de documentos como resposta a pedidos de clientes *Web*. Foi usada por permitir que, código *Java* e certas acções pré-definidas fiquem incorporadas num contexto de página estática. Cada pedido feito por uma página, é compilado em tempo-real no servidor. Sendo que também é um método mais fácil e mais rápido de mostrar o conteúdo dos parâmetros [JSP, 2011].

O *Struts 1.3.8* é uma ferramenta de aplicações *Web open-source* para o desenvolvimento de aplicações *Java EE*. Usa e alarga o *Java Servlet API* para encorajar os programadores a adoptar o modelo *Model-View-Controller* (MVC). Este modelo consiste em separar os três componentes da metodologia, isto é, separa o *Model*¹, a *View*² e o *Controller*³. [?].

¹Componente de interacção com a base de dados

²Visualização por parte do utilizador na página *web*

³Controlo feito pelo sistema

A troca de informações efectuada entre a *Camada de Interface* e a *Camada de Controlo*, é feita através de *Webservices* e os pacotes trocados são em XML. O XML foi originalmente concebido para publicações electrónicas de grande escala, na actualidade, desempenha um papel cada vez mais importante na troca de dados na *Web*. O facto de ser uma derivação do *Standard* (ISO 8879), ou seja, é uma linguagem *Standard* importante, pois é um dos requisitos do projecto [XML, 2011].

A ferramenta *Netbeans* é uma plataforma para aplicações *Desktop* em *Java*, é ainda um IDE que permite desenvolver aplicações com *Java*, *JavaScript*, *PHP*, *C*, *C++* entre outros. A escolha deste IDE foi feita pelo facto deste permitir que sejam criadas páginas *Web*, aplicações *Desktop* e aplicações móveis utilizando a plataforma *Java*, bem como outras linguagens de programação como por exemplo o *JavaFx* e o *PHP*[NET, 2011].

O *Hibernate* é uma biblioteca do tipo *Object/Relational Mapping* (ORM) para a linguagem *Java*, fornecendo uma forma mais fácil para o mapeamento de estruturas orientadas a objectos para as tradicionais bases de dados. Optou-se por utilizar esta tecnologia devido à maior facilidade na passagem das classes *Java* para tabelas de base de dados e de tipos de dados *Java*, para tipo de dados *Structured Query Language* (SQL), visto ser essa a sua principal característica. Possibilita ainda uma maior simplicidade nas consultas e recuperação de dados das bases de dados [HIB, 2011].

A tecnologia usada para a gestão das bases de dados do sistema foi o *MySQL*, que é um sistema de administração de bases de dados e funciona como um servidor permitindo um acesso multi-utilizador a um número indeterminado de bases de dados. O recurso a esta ferramenta foi feito pelo facto de ser uma aplicação de bases de dados sem custos, com alta fiabilidade e com uma performance elevada [SQL, 2011].

Como tecnologia usada para elaborar os diagramas *Unified Modeling Language* (UML), optou-se por usar o *Visual Paradigm for UML*, que é uma ferramenta que suporta diagramas UML e Business Process Modeling Notation (BPMN). Fornece modelação empresarial e um criador de ORM para *Java*, *.NET* e *PHP*. Foi escolhido o *Visual Paradigm for UML* devido a possuir um leque muito vasto de diagramas UML [UML, 2011].

5.2 Visão Estrutural

Nesta secção estará representada uma visão estrutural do sistema, modelada através de diagramas UML, que permitirá uma maior compreensão do mesmo.

5.2.1 Diagramas de Classes

Nesta subsecção serão retratados os diagramas UML de classes dos diversos processos constituintes deste projecto.

5.2.1.1 Camada de Interface do NEMO-M

Os diagramas de classe que constituem esta camada, correspondem a classes que pertencem ao NEMO-P.

A classe "*RegisterForm*" apresentada na Figura 5.1, representa a classe que recebe os dados por parte do utilizador para que este se possa registar no sistema NEMO-M.

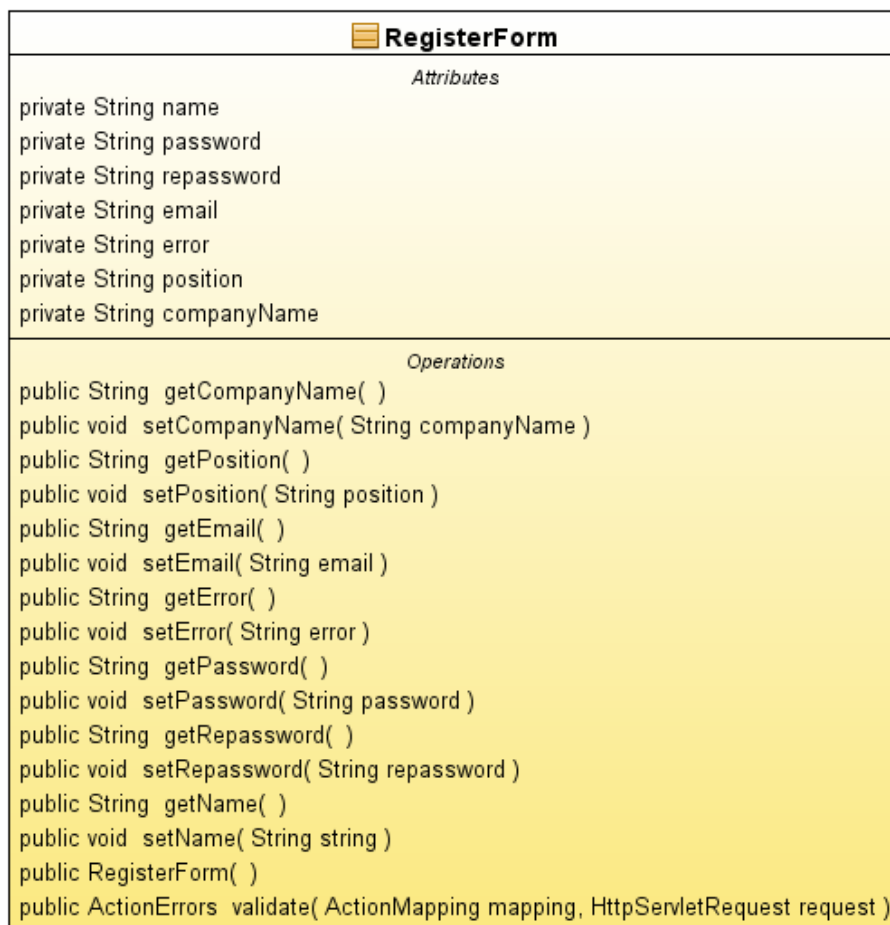


Figura 5.1: Diagrama da Classe *RegisterForm*

5.2.1.2 Camada de Controlo do NEMO-M

Controlo dos Dados Inseridos no NEMO-P

Os processos do sistema que controlam os pedidos feitos pelo utilizador são semelhantes em todos os casos, por isso optou-se por mostrar apenas uma classe que representa esse processo.

A classe "*RegisterAction*", ilustrada na Figura 5.2, representa o controle existente para cada dado inserido pelo utilizador.

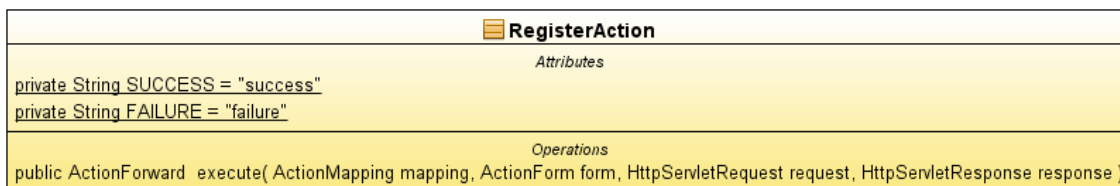


Figura 5.2: Diagrama da Classe *RegisterAction*

IED Configuration Manager

Um dos processos elaborados foi o responsável pela administração da base de dados dos IEDs. Cujos diagramas de classes serão apresentados e explicados em seguida.

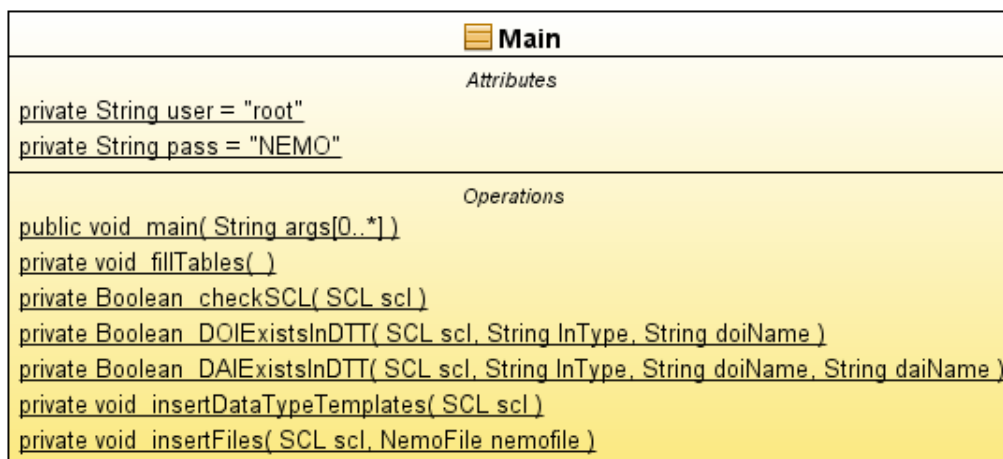
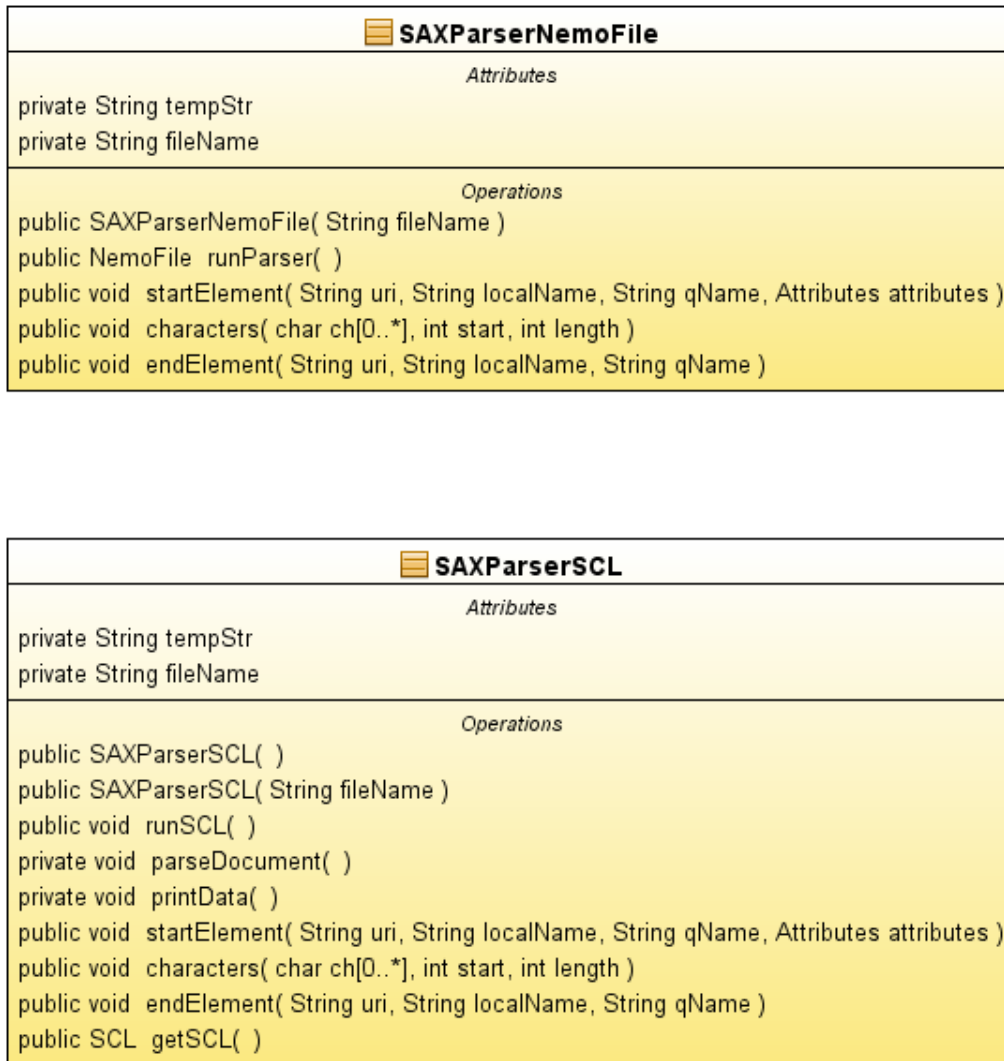


Figura 5.3: Diagrama da Classe Principal desta Componente

Na Figura 5.3 está representada a classe principal do módulo responsável pelo tratamento dos IEDs existentes no sistema. Assim, esta classe tem como função administrar todo o processo de leitura e armazenamento das informações contidas nos ficheiros recebidos pelo NEMO-M. Para isso, esta classe dispõe de diversas classes auxiliares que ajudaram no procedimento a seguir, classes essas que serão representadas e descritas em seguida.

As classes representadas na Figura 5.4, de seu nome, "*SAXParserNemoFile*" e "*SAXParserSCL*", são classes auxiliares responsáveis pela análise da informação proveniente dos dois ficheiros, o NEMO-F e o ficheiro SCL. Posteriormente colocam esses dados nas estruturas criadas, destinados para tal. As classes têm a capacidade de passar da linguagem XML, linguagem em que os ficheiros estão escritos, para a linguagem de programação utilizada, neste caso o *JAVA*.

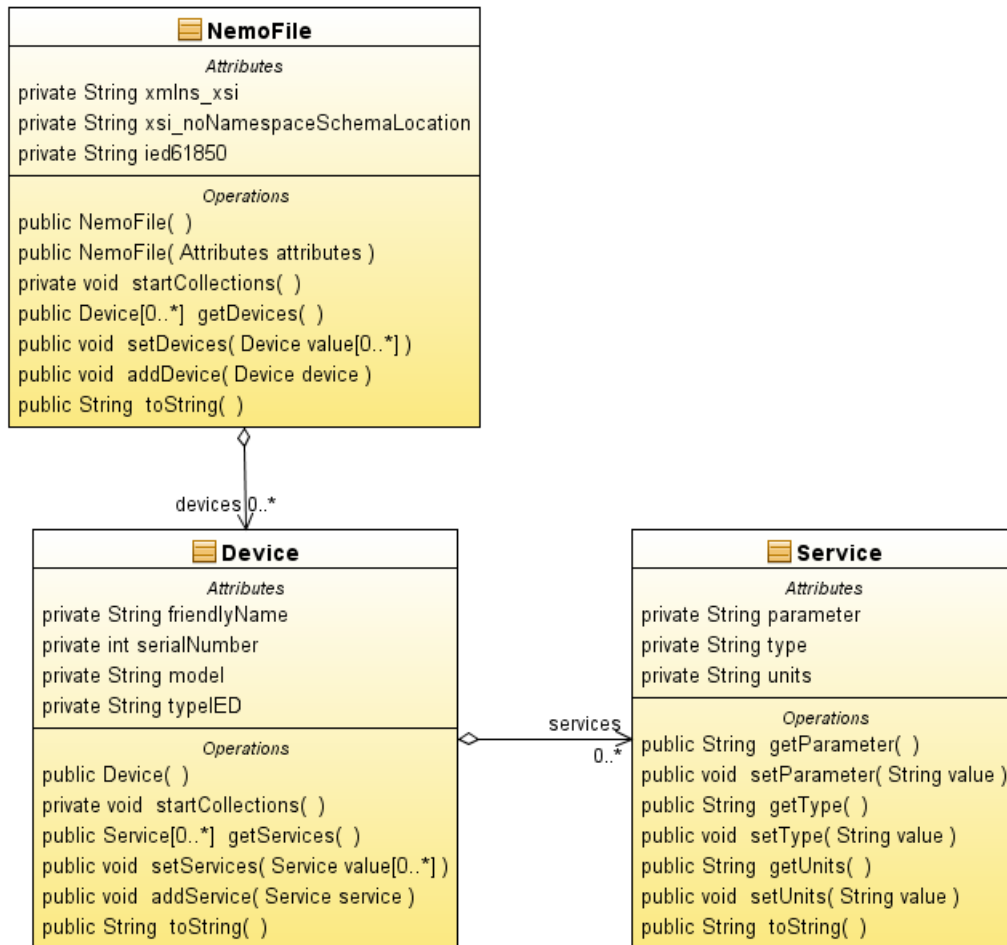
Figura 5.4: Diagramas das Classes *Parsers*

A classe representada na Figura 5.5 corresponde a uma estrutura auxiliar criada para armazenar a informação proveniente do NEMO-F. Nas classes "*NemoFile*" e "*Device*" faltam os métodos *GET* e *SET* para todos os atributos das mesmas, que não estão representados para maior percepção da figura.

Da mesma maneira, foi criada uma estrutura auxiliar para guardar os dados que advém do ficheiro SCL. Optou-se por não disponibilizar a figura, visto tratar-se de uma imagem de difícil visualização e por ser semelhante ao DER representado na secção 4.4.

NEMO Service Manager (Internal Services)

Este processo é o responsável por gerir a base de dados dos *Services*, processo que foi explicado em detalhe na secção 4.3, para o efeito foram criadas diversas classes, cujos diagramas de classe UML

Figura 5.5: Diagrama da Classe *NemoFile*

serão ilustrados em seguida.

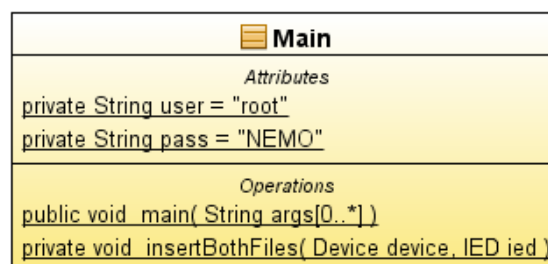


Figura 5.6: Diagrama da Classe Principal desta Componente

A Figura 5.6 representa o diagrama de classes UML da classe que administra todo o processo de procura de novos serviços para disponibilizar no sistema, colocando-os na base de dados dos *Services*. Para o bom funcionamento do processo, esta classe usou diversas classes que foram criadas para auxiliar e que serão referidas de seguida.

Foram criadas diversas estruturas para armazenar os dados temporariamente durante o processo que

são:

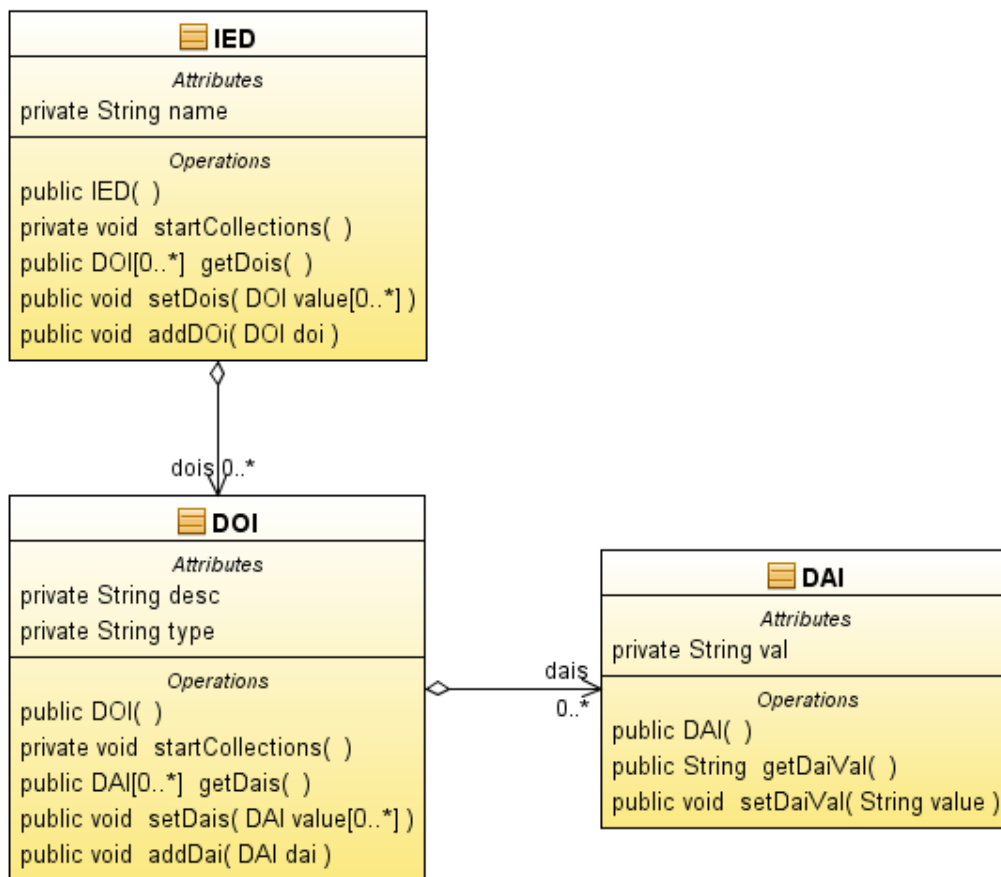


Figura 5.7: Diagrama da Classe do tipo *IED*

As classes ilustradas na Figura 5.7 representam a estrutura que guarda as informações sobre os IEDs na base de dados, armazenando os *Services* que advém do ficheiro SCL.

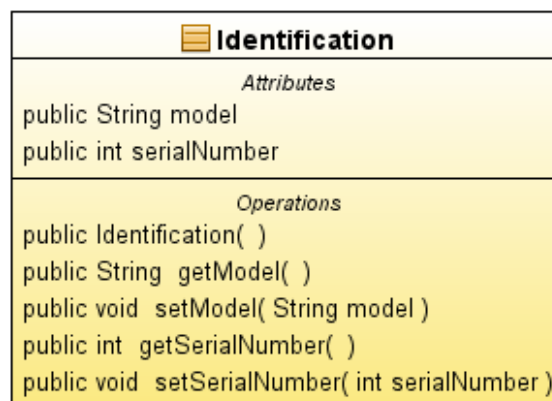


Figura 5.8: Diagrama da Classe do tipo *Identification*

Visto o ID dos IEDs ser uma chave primária dupla, foi criada a estrutura "*Identification*", cuja classe

está representada na Figura 5.8, que foi usada para a procura de novos *IEDs* no sistema, retornando a sua identificação nesta variável, procura essa feita na base de dados dos *IEDs*.

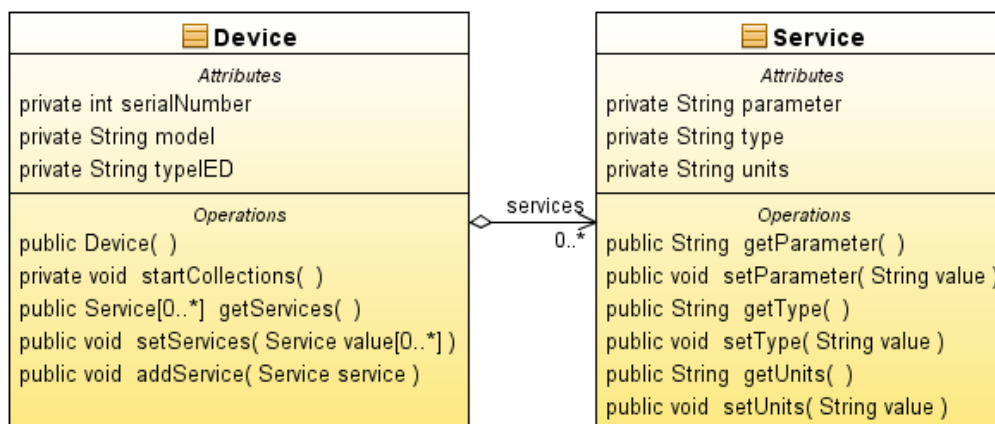


Figura 5.9: Diagrama da Classe do tipo *Device*

Por último, foi criada uma estrutura para armazenar os serviços provenientes do NEMO-F, as classes que se encontram na Figura 5.9 representam a variável em questão.

5.2.1.3 Camada de Entidade do NEMO-M

Nesta camada encontram-se as classes que fazem as ligações às bases de dados do sistema NEMO-M, como já fora visto na Figura 4.5.

A classe "*Inserts*" representada na Figura 5.10 a classe "*QuerysIEDs*" que se encontra na Figura 5.11, ambas são classes que servem de intermediária entre a Camada de Controlo e a base de dados dos *IEDs*.

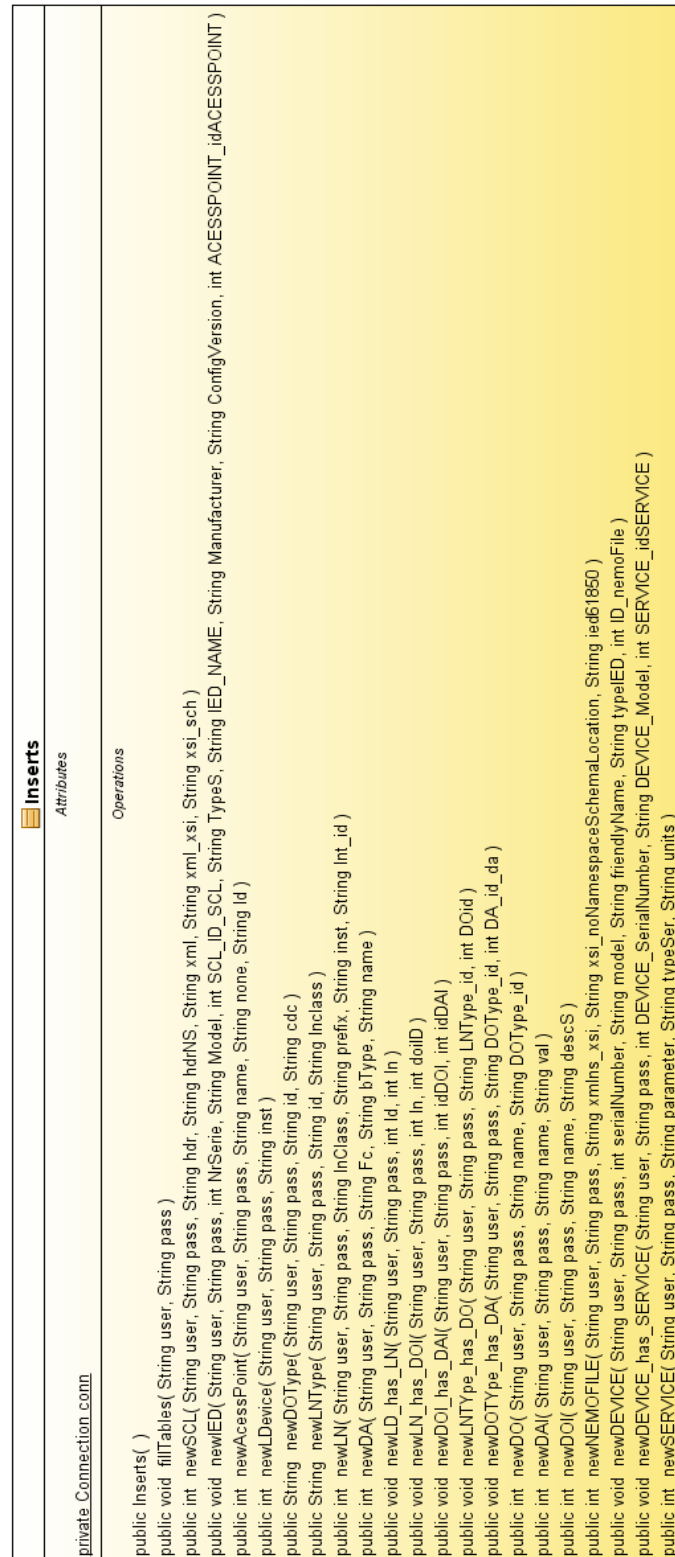
As Figuras 5.12 e 5.13 representam respectivamente as classes "*InsertsServices*" e "*QuerysServices*" e constituem uma parte da camada de entidade que permite ligações à base de dados dos *Services*.

A classe "*ServiceUtils*", Figura 5.14, é usada como intermediária entre a camada de controlo que controla os dados inseridos pelos utilizadores do NEMO-M e a base de dados dos *Services*.

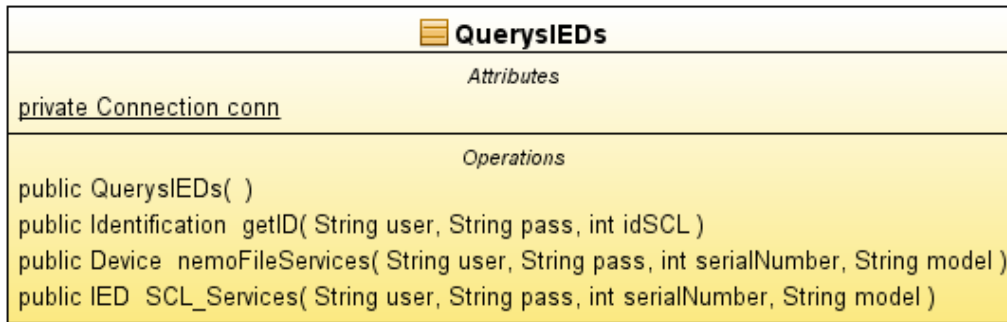
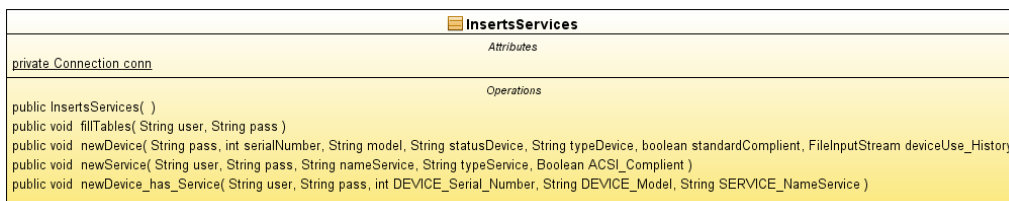
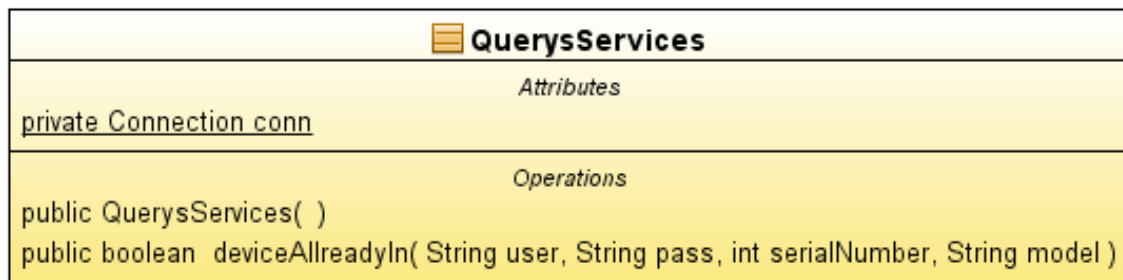
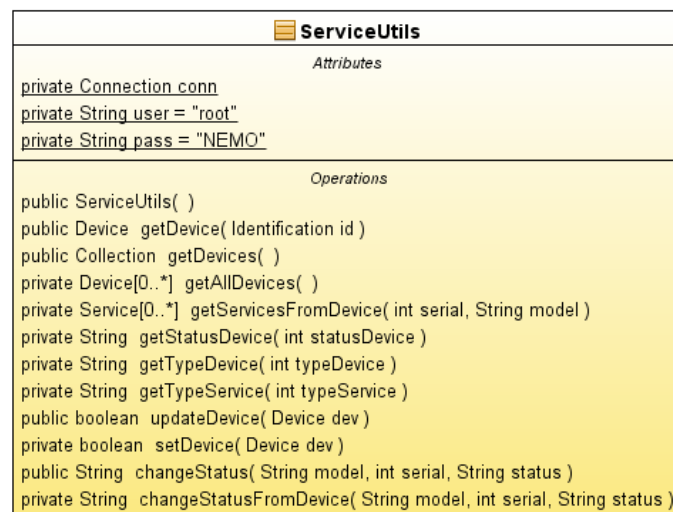
Na Figura 5.15 encontra-se a classe "*Utils*" que faz as ligações com a base de dados dos *Users*.

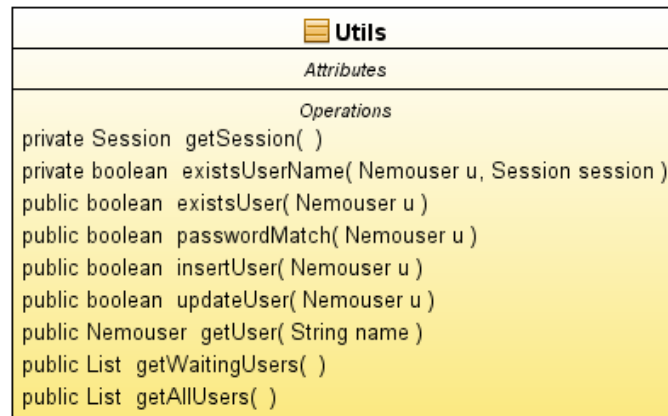
5.2.2 Diagramas de Sequência

Nesta subsecção serão apresentados os diagramas de sequência de processos integrantes do NEMO-M, não cumprem totalmente o *Standard* dos diagramas de sequência, mas assim ilustram da melhor maneira possível a situação em questão. Do elevado número de casos de uso, resulta o mesmo número

Figura 5.10: Diagrama da Classe *Inserts*

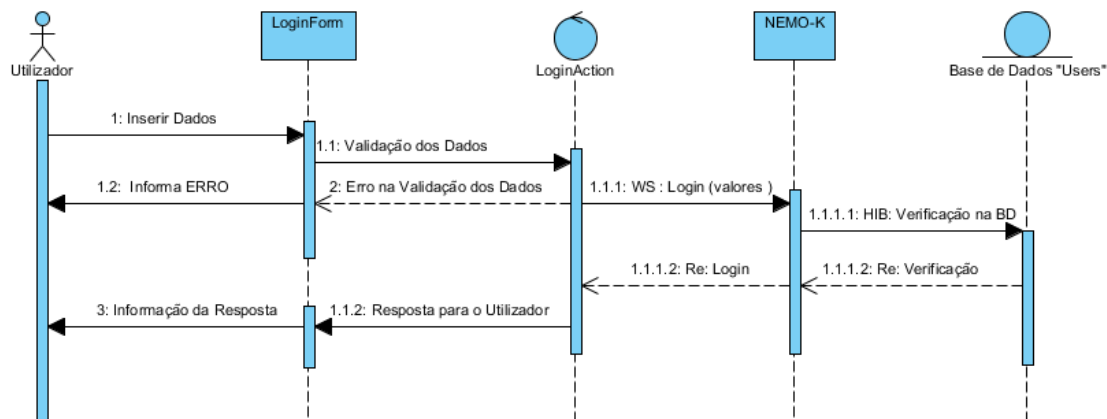
de diagramas de sequência. Sendo assim, considerou-se excessivo a representação de todos, optando apenas por ilustrar alguns exemplos. Nesta análise aos diagramas, optou-se ainda por representar dois processos internos do trabalho elaborado.


Figura 5.11: Diagrama da Classe *QuerysIEDs*

Figura 5.12: Diagrama da Classe *Inserts*

Figura 5.13: Diagrama da Classe *QuerysServices*

Figura 5.14: Diagrama da Classe *ServicesUtils*

Figura 5.15: Diagrama da Classe *Utils*

5.2.2.1 Funcionalidade *Login*

Na Figura 5.16 encontra-se representado o diagrama de sequência do processo de iniciar sessão de um utilizador no sistema NEMO-M.

Figura 5.16: Diagrama de Sequência da funcionalidade *Login*

Em primeiro lugar, o utilizador insere os dados do *Login*: *Username* e *Password*, estes dados são recebidos na *LoginForm*, que por sua vez os validará com o auxílio da Camada de Controlo da mesma *LoginAction*. Em caso de alguma anomalia dos mesmos o processo é interrompido, informando o utilizador do ERRO ocorrido, caso contrário, os dados são enviados através de um *Web Service* disponibilizado pelo NEMO-K para que este possa proceder à validação dos mesmos. Para isso, o NEMO-K faz a pesquisa na base de dados dos *Users*, obtendo a resposta sobre a mesma, que será reencaminhada para o utilizador, ficando assim a saber se o *Login* foi feito, ou não, com sucesso.

5.2.2.2 Funcionalidade *AcceptUser*

A Figura 5.17 representa o processo que o administrador do sistema tem que efectuar para permitir, ou negar, o acesso ao sistema NEMO-M por parte de outros utilizadores.

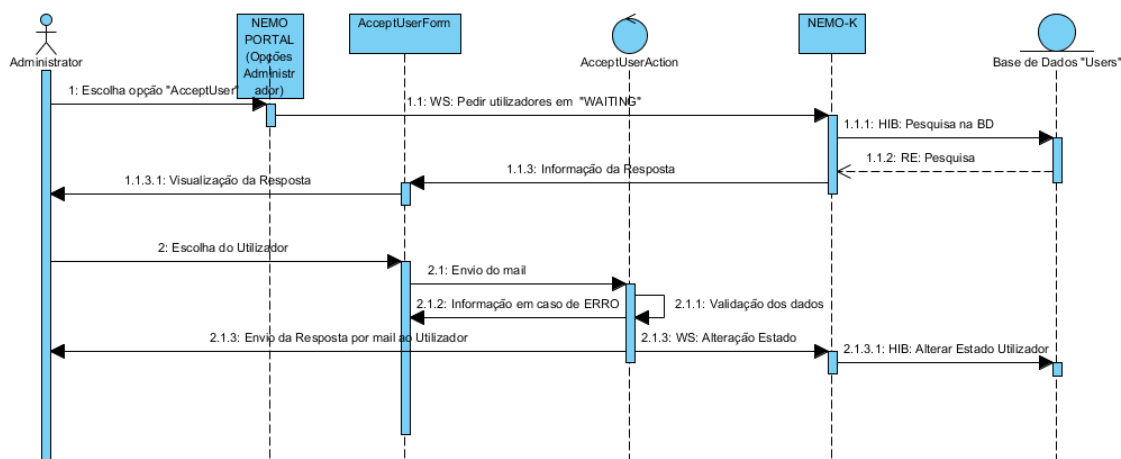


Figura 5.17: Diagrama de Sequência da funcionalidade *AcceptUser*

O administrador do sistema ao escolher a opção *AcceptUser*, o NEMO-P acede ao *Web Service* do NEMO-K que após proceder a uma pesquisa na base de dados dos *Users*, disponibiliza uma página *Web* com uma lista dos utilizadores que estão em estado de espera, caso eles existam. A partir daí o administrador tem a possibilidade de analisar o pedido para novo membro do sistema por parte de cada utilizador. Para isso, o administrador procede ao preenchimento dos campos existentes na página *AcceptUserForm* e de seguida escolhe a opção para enviar por mail a resposta ao utilizador. O *AcceptUserAction* procede à validação dos dados inseridos, caso exista alguma inconformidade no preenchimento dos campos o administrador é informado e esta parte do processo interrompida. Caso os dados sejam validados positivamente é enviado um mail ao utilizador com a resposta, ao mesmo tempo que é alterado o campo "STATUS" na base de dados *Users* para o valor que o administrador definir.

5.2.2.3 Funcionalidade *Inserção de um IED*

O processo pelo qual um IED passa ao ser inserido na base de dados dos *IEDs*, está ilustrado na Figura 5.18, que passará a ser descrito em seguida com maior pormenor.

Este processo não advém dos Casos de Uso já referidos, é um processo invisível ao olho do utilizador, mas deveras importante para o funcionamento da NEMO-M. Consiste inicialmente na descoberta de novos dispositivos na rede do sistema. Após a qual os ficheiros provenientes dos dispositivos são processados nesta funcionalidade. Em primeiro lugar as informações provenientes do ficheiros são passada para estruturas auxiliares, processo feito no *IED Configuration Manager*. Em seguida, é feita uma consulta à base de dados dos *IEDs* para verificar a possível existência do mesmo no sistema. Caso não exista, procede-se à inserção de todos os dados referentes ao IED na base de dados, pas-

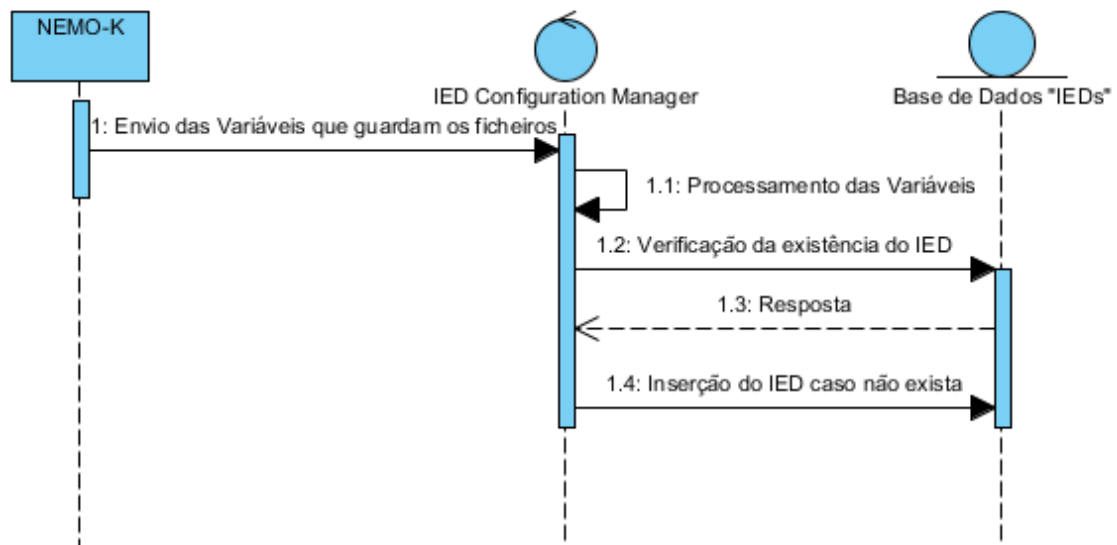


Figura 5.18: Diagrama de Sequência da funcionalidade *Inserção dum IED*

sando assim a estar disponível para uso no sistema.

5.2.2.4 Funcionalidade *Inserção dos Serviços*

A Figura 5.19 representa o processo que o NEMO-M executa para descobrir novos serviços disponíveis e de seguida proceder à inserção dos mesmos na base de dados dos *Services*.

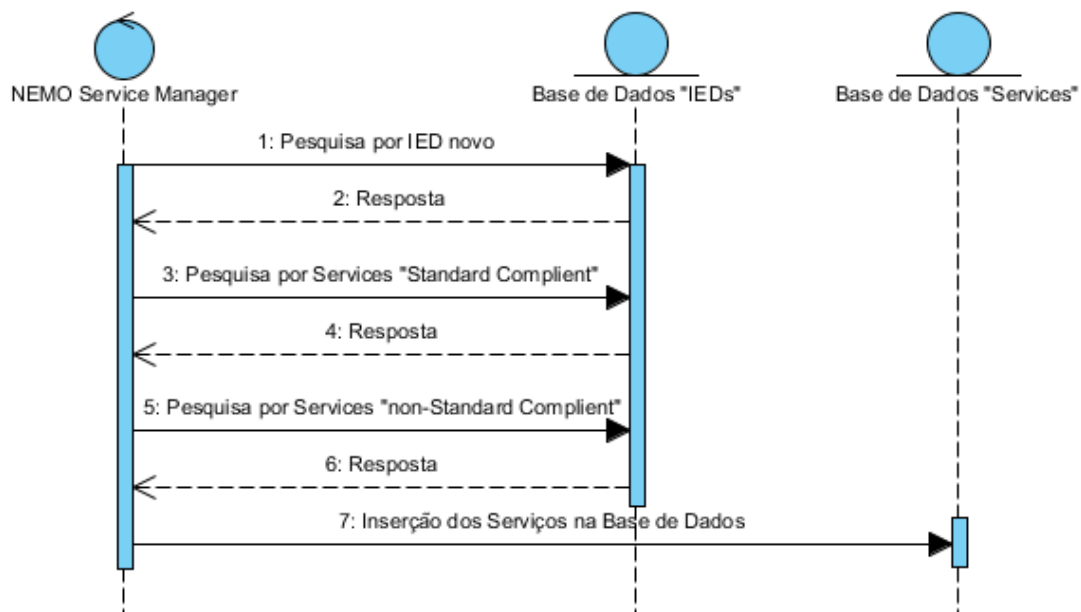


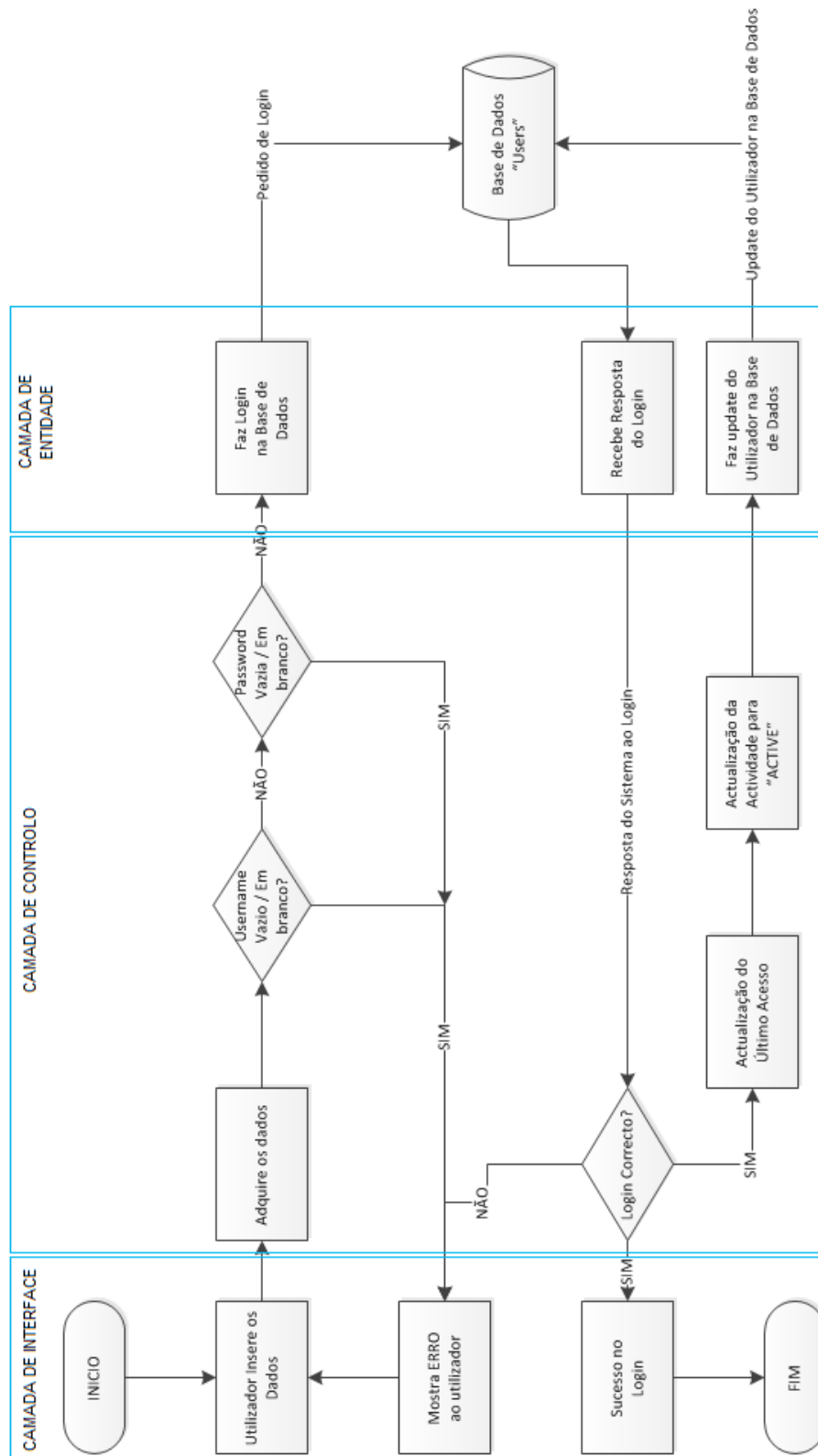
Figura 5.19: Diagrama de Sequência da funcionalidade *Manage Services*

Como último exemplo de um diagrama de sequência do sistema tem-se o diagrama que caracteriza o processo de administração dos serviços que podem ser usados no sistema. Em primeiro lugar é feita uma pesquisa por parte do *NEMO Service Manager* por um novo IED na base de dados dos mesmos, caso exista, procede-se à procura dos serviços que o dispositivo tem. É feita uma pesquisa individual aos serviços "*Standard Compliant*" e "*non-Standard Compliant*" guardando-os em variáveis auxiliares distintas. Por fim, as informações contidas nessas variáveis são inseridas na base de dados dos *Services* associadas ao IED em questão.

5.2.3 Diagramas de Actividade

Nesta subsecção será representado um exemplo de um diagrama de actividades relativo a uma das funcionalidades existentes no sistema.

Assim, a Figura 5.20 representa o diagrama de actividades do mecanismo que possibilita um utilizador executar o *Login* no sistema. Esta funcionalidade que fora previamente ilustrada e descrita na Figura 5.16, é agora descrita num sentido das actividades efectuadas para o seu bom funcionamento. Este diagrama de actividades demonstra o fluxo pelo qual o processo segue para chegar ao seu fim. Na Figura 5.20 pode-se constatar que o processo tem início com o utilizador a inserir os dados referentes ao início de sessão num sistema, isto é, colocando o *Username* e a *Password* na página *Web* dedicada para tal. De seguida, esses dados são adquiridos pela *Camada de Controlo* que processa as duas variáveis uma a uma, informando o utilizador em caso de erro, caso contrário, procede à tentativa de *Login* no sistema com a consulta à base de dados dos *Users*. A resposta a essa tentativa é novamente enviada à *Camada de Controlo* que faz a validação da mesma. Caso o *Login* seja bem executado as variáveis "*LastAccess*" e "*Activity*" são actualizadas para os novos valores e por sua vez alteradas na base de dados, o utilizador é também informado sobre o sucesso do mesmo, sendo redireccionado para nova página *Web*, dando assim fim ao processo.

Figura 5.20: Diagrama de Atividades da funcionalidade *Login*

5.3 Exemplo de Utilização

Como exemplo de utilização do NEMO-M irá ser mostrado a inserção de um novo IED na rede do sistema e de seguida irá ser mudado o estado de um dos dispositivos que encontram na rede.



Nemo Login!

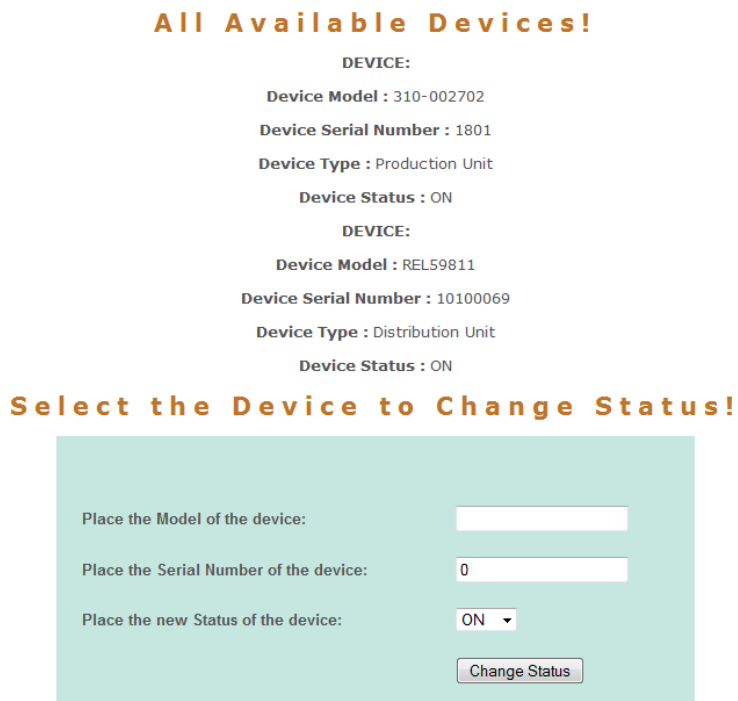
Enter your name:

Enter your password:

Other Options !

Figura 5.21: Página Inicial do NEMO-M

O administrador do NEMO-M depois de fazer "Login" na opção que se pode visualizar na Figura 5.21, pode escolher a funcionalidade "Change Device Status" que lhe permite ver todos os IEDs que se encontram na rede do sistema NEMO-M, Figura 5.22, neste caso apenas dois dispositivos.



All Available Devices!

DEVICE:

Device Model : 310-002702

Device Serial Number : 1801

Device Type : Production Unit

Device Status : ON

DEVICE:

Device Model : REL59811

Device Serial Number : 10100069

Device Type : Distribution Unit

Device Status : ON

Select the Device to Change Status!

Place the Model of the device:

Place the Serial Number of the device:

Place the new Status of the device:

Figura 5.22: Dispositivos na Rede do NEMO-M

Considerando que entrou um novo dispositivo na rede, ele será automaticamente inserido na base de dados dos *IEDs* e por sua vez na dos *Services*, logo quando o administrador voltar a aceder a esta funcionalidade irá encontrar três dispositivos na rede, como se pode ver na Figura 5.23, com o novo dispositivo em evidência.

All Available Devices!

DEVICE:

Device Model : 310-002702
Device Serial Number : 1801
Device Type : Production Unit
Device Status : ON

DEVICE:

Device Model : REL59811
Device Serial Number : 10100069
Device Type : Distribution Unit
Device Status : ON

DEVICE:

Device Model : FD3.6-2000-10L
Device Serial Number : 36200010
Device Type : Production Unit
Device Status : ON

Select the Device to Change Status!

Place the Model of the device:

Place the Serial Number of the device:

Place the new Status of the device:

Figura 5.23: Dispositivo Novo na Rede do NEMO-M

O administrar do NEMO-M pode agora optar por mudar o estado de um dos dispositivos de produção que se encontram na rede do sistema, neste exemplo optou-se por mudar o dispositivo de produção que já se encontrava na rede, preenchendo os campos para tal, como demonstra a Figura 5.23 e carregando em *"Change Status"*.

Após feito o pedido de alteração, se este for bem sucedido como no exemplo em questão aparecerá a mesma página *web* com a mensagem que se pode ver a vermelho na Figura 5.24 "Successful Update!" e o IED com o estado alterado.

All Available Devices!

DEVICE:

Device Model : 310-002702
Device Serial Number : 1801
Device Type : Production Unit
Device Status : OFF

DEVICE:

Device Model : REL59811
Device Serial Number : 10100069
Device Type : Distribution Unit
Device Status : ON

DEVICE:

Device Model : FD3.6-2000-10L
Device Serial Number : 36200010
Device Type : Production Unit
Device Status : ON

Select the Device to Change Status!

Successful Update !

Place the Model of the device:

Place the Serial Number of the device:

Place the new Status of the device:

Figura 5.24: Estado do Dispositivo Alterado

Capítulo 6

Validação

Este capítulo tem como objectivo mostrar ao leitor o sistema em funcionamento, ilustrando a sequência de processo pelo qual cada um dos componentes do NEMO-M passa.

6.1 *IED Configuration Manager*

Este módulo entra em funcionamento após a entrada de um novo dispositivo na rede. Este módulo depois de receber os ficheiros provenientes de um novo dispositivo, processa toda a sua informação e armazena-a na base de dados dos *IEDs*, como se verifica na Figura 6.1.

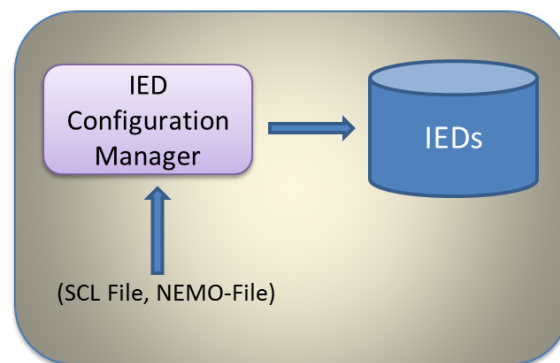


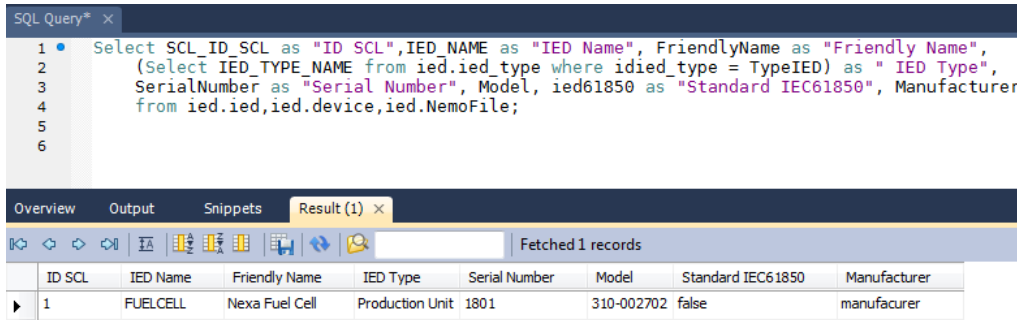
Figura 6.1: Funcionamento do módulo *IED Configuration Manager*

6.1.1 Exemplo em Funcionamento

Supondo que já existe um dispositivo na rede do NEMO-M, fez-se uma primeira *Query*¹ à base de dados que permitiu verificar este facto, como se pode ver na Figura 6.2.

¹Pesquisa a uma base de dados.

6.1. IED CONFIGURATION MANAGER



The screenshot shows a SQL query window with a query that selects IED configuration data. Below the query, the 'Result (1)' tab shows a table with 1 record.

```
SQL Query* X
1 Select SCL_ID_SCL as "ID SCL", IED_NAME as "IED Name", FriendlyName as "Friendly Name",
2 (Select IED_TYPE_NAME from ied.ied_type where idied_type = TypeIED) as " IED Type",
3 SerialNumber as "Serial Number", Model, ied61850 as "Standard IEC61850", Manufacturer
4 from ied.ied, ied.device, ied.NemoFile;
5
6
```

ID SCL	IED Name	Friendly Name	IED Type	Serial Number	Model	Standard IEC61850	Manufacturer
1	FUELCELL	Nexa Fuel Cell	Production Unit	1801	310-002702	false	manufacurer

Figura 6.2: Primeira Query à base de dados dos IEDs.

Neste caso o dispositivo é o "Fuel Cell", mostrando uma série de propriedades do mesmo que estavam contidas nos ficheiros recebidos.

Simulando a entrada de um novo dispositivo na rede, neste caso o "Aerogerador", as Figuras 6.3 e 6.4 representam respectivamente excertos do *SCL File* e do *NEMO-File* provenientes do novo IED que entrou no rede do NEMO-M.

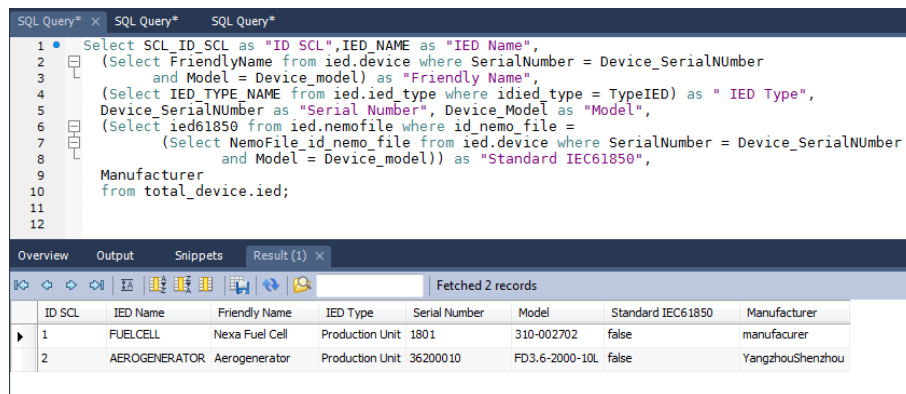


The screenshot shows an XML file named 'SCLFileAerogenerator.xml'. The XML content defines the configuration for an aerogenerator (wind turbine) IED, including its name, type, manufacturer, and various logical nodes (LN) for monitoring and control.

```
<?xml-stylesheet type="text/xsl" href="SCL.xsl"?>
<SCL xmlns="http://www.iec.ch/61850/2003/SCL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.iec.ch/61850/2003/SCL SCL.xsd">
  <Header id="BCU" nameStructure="AEROGENERATOR"/>
  <!-- Header Elements listed here (IP address, MAC address, etc) -->
  <IED name="AEROGENERATOR" type="Production Unit" manufacturer="YangzhouShenzhou" configVersion="1.0">
    <Services>
      <!-- List Services here (Logging, GOOSE max, FileHandling, etc) -->
    </Services>
    <AccessPoint name="P1">
      <Server>
        <Authentication none="true" />
        <LDevice inst="U1">
          <!-- Other Logical Nodes -->
          <LN lnType="MMON_0" lnClass="MMON" prefix="" inst="1">
            <DOI name="Vol" desc="SystemVoltage"/>
            <DOI name="Amp" desc="SystemCurrent"/>
            <DOI name="Watt" desc="SystemActivePower"/>
            <DOI name="VoltAmp" desc="SystemReactivePower"/>
            <DOI name="VoltAmp" desc="SystemApparentPower"/>
            <DOI name="PwrFact" desc="SystemPowerFactor"/>
            <DOI name="Hz" desc="Frequency"/>
          </LN>
          <LN lnType="MMOU_0" lnClass="MMOU" prefix="" inst="1">
            <DOI name="TotM" desc="SystemActivePower"/>
            <DOI name="TotVar" desc="SystemReactivePower"/>
            <DOI name="TotVA" desc="SystemApparentPower"/>
            <DOI name="TotPF" desc="SystemPowerFactor"/>
            <DOI name="Hz" desc="Frequency"/>
          </LN>
          <LN lnType="MMTR_0" lnClass="MMTR" prefix="" inst="1">
            <DOI name="SupWh" desc="ImportedActiveEnergy"/>
            <DOI name="SupVarh" desc="ImportedReactiveLaggingEnergy"/>
            <DOI name="DmdWh" desc="ExportedActiveEnergy"/>
            <DOI name="DmdVarh" desc="ExportedReactiveLaggingEnergy"/>
          </LN>
        </LDevice>
      </Server>
    </AccessPoint>
  </IED>
</SCL>
```

Figura 6.3: Excerto do *SCL File* do "Aerogerador".

Após o processamento de ambos os ficheiros por parte do *IED Configuration Manager* e efectuando nova Query à base de dados, pode-se verificar que as informações sobre este novo dispositivo já se encontram armazenadas, como ilustra a Figura 6.5.

Figura 6.4: Excerto do *NEMO-File* do "Aerogerador".Figura 6.5: Segunda *Query* à base de dados dos IEDs.

6.2 NEMO Service Manager (Internal Services)

Este módulo é processado logo após o módulo *IED Configuration Manager* terminar. O seu funcionamento está representado na Figura 6.6, e consiste em verificar na base de dados dos *IEDs* a informação que acabou de ser inserida pelo módulo *IED Configuration Manager* e após retirar e tratar a informação necessária, armazena-a na base de dados dos *Services*.

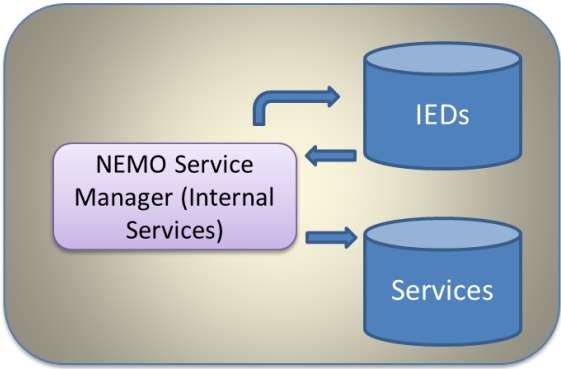


Figura 6.6: Funcionamento do módulo *NEMO Service Manager (Internal Services)*

6.2.1 Exemplo em Funcionamento

Considerando o mesmo exemplo usado para o módulo *IED Configuration Manager*, a Figura 6.7 re-
presenta uma *Query* à base de dados quando apenas se encontra na rede o "Fuel Cell".

The screenshot shows an SQL query interface with a query editor and a results table. The query is as follows:

```
1 Select Serial_Number as "Serial Number", Model,  
2 (Select statusname from services.status_device where statusDevice = idStatus) as "Status",  
3 (Select Type_Device from services.type_device where typeDevice = idType_Device) as "Type",  
4 Service_NameService as "Services"  
5 from services.device, services.device_has_service  
6 where Device_serial_Number = Serial_Number and Device_model = Model;  
7  
8
```

The results table shows 17 records. The columns are Serial Number, Model, Status, Type, and Services. The data is as follows:

Serial Number	Model	Status	Type	Services
1801	310-002702	ON	Production Unit	AirPumpOperatingVoltage
1801	310-002702	ON	Production Unit	AirTemperature
1801	310-002702	ON	Production Unit	AmbientTemperature
1801	310-002702	ON	Production Unit	BatteryVoltage
1801	310-002702	ON	Production Unit	BridgeVoltage
1801	310-002702	ON	Production Unit	CumulativeHydrogenConsumption
1801	310-002702	ON	Production Unit	FailCode
1801	310-002702	ON	Production Unit	FuelPressure
1801	310-002702	ON	Production Unit	HydrogenConcentration
1801	310-002702	ON	Production Unit	HydrogenPressure
1801	310-002702	ON	Production Unit	OxygenConcentration
1801	310-002702	ON	Production Unit	PurgeCellVoltage
1801	310-002702	ON	Production Unit	StackCurrent
1801	310-002702	ON	Production Unit	StackTemperature
1801	310-002702	ON	Production Unit	StackVoltage
1801	310-002702	ON	Production Unit	Status
1801	310-002702	ON	Production Unit	WarningBitmap

Figura 6.7: Primeira *Query* à base de dados dos Services.

Após a entrada do "Aerogerador" e o devido processamento por parte deste módulo, a base de dados dos *Services* fica preenchida com as entradas correspondentes ao novo dispositivo e todos os seus serviços. Essa informação encontra-se representada na Figura 6.8.

The screenshot shows a SQL query editor with a query that selects service details for a specific device. The query is as follows:

```

1 Select Serial_Number as "Serial Number", Model,
2   (Select statusname from services.status_device where statusDevice = idStatus) as "Status",
3   (Select Type_Device from services.type_device where typeDevice = idType_Device) as "Type",
4   Service_NameService as "Services"
5 from services.device, services.device_has_service
6 where Device_serial_Number = Serial_Number and Device_model = Model;

```

Below the query editor, the 'Result (1)' tab is active, showing a table with 28 records. The table has the following columns: Serial Number, Model, Status, Type, and Services. The data is organized into two groups of 12 records each, separated by a blank row.

Serial Number	Model	Status	Type	Services
1801	310-002702	ON	Production Unit	ComulativeHydrogenConsumption
1801	310-002702	ON	Production Unit	FailCode
1801	310-002702	ON	Production Unit	FuelPressure
1801	310-002702	ON	Production Unit	HydrogenConcentration
1801	310-002702	ON	Production Unit	HydrogenPressure
1801	310-002702	ON	Production Unit	OxygenConcentration
1801	310-002702	ON	Production Unit	PurgeCellVoltage
1801	310-002702	ON	Production Unit	StackCurrent
1801	310-002702	ON	Production Unit	StackTemperature
1801	310-002702	ON	Production Unit	StackVoltage
1801	310-002702	ON	Production Unit	Status
1801	310-002702	ON	Production Unit	WarningBitmap
36200010	FD3.6-2000-10L	ON	Production Unit	DemandIntegrationTime
36200010	FD3.6-2000-10L	ON	Production Unit	EnergyCounterReset
36200010	FD3.6-2000-10L	ON	Production Unit	FirmwareVersionNumber
36200010	FD3.6-2000-10L	ON	Production Unit	Frequency
36200010	FD3.6-2000-10L	ON	Production Unit	InstrumentModel
36200010	FD3.6-2000-10L	ON	Production Unit	PeakOffNeutralCurrent
36200010	FD3.6-2000-10L	ON	Production Unit	PeakOffSystemActivePowerDemand
36200010	FD3.6-2000-10L	ON	Production Unit	PeakOffSystemApparentPowerD...
36200010	FD3.6-2000-10L	ON	Production Unit	PeakReset
36200010	FD3.6-2000-10L	ON	Production Unit	SerialNumber
36200010	FD3.6-2000-10L	ON	Production Unit	StackCurrent

Figura 6.8: Segunda Query à base de dados dos Services.

6.3 NEMO External Services

De modo a permitir o acesso por parte do NEMO-P às informações contidas nas diversas bases de dados foram criados dois *Web Services*. Um *Web Service* para comunicar com a base de dados dos *Services*, Figura 6.9. Outro *Web Service* para trocar informações com a base de dados dos *Users*, Figura 6.10.

O *Web Service* da Figura 6.9 contém os seguintes *Web Methods*:

- **getAllDevices:** Retorna toda a informação armazenada na base de dados dos *Services* todos os dispositivos existentes no sistema.
- **changeStatusDevice:** Usado para mudar o estado de um dispositivo, indicando o seu ID, neste caso o Número de Série e o Modelo do dispositivo.

O *Web Service* da Figura 6.10 contém os seguintes *Web Methods*:

- **register:** Insere um novo utilizador na base de dados dos *Users*.

NemoServicesServicesService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.util.List com.nemo.webservices.services.NemoServicesServices.getAllDevices()

()

public abstract java.lang.String com.nemo.webservices.services.NemoServicesServices.changeStatusDevice(java.lang.String,int,java.lang.String)

(, ,)

Figura 6.9: Métodos do *Web Service* que interage com a base de dados dos *Services*.

NemoUsersServicesService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.String com.nemo.webservices.users.NemoUsersServices.register(com.nemo.webservices.users.Nemouser)

()

public abstract com.nemo.webservices.users.Nemouser com.nemo.webservices.users.NemoUsersServices.getUser(java.lang.String)

()

public abstract java.lang.String com.nemo.webservices.users.NemoUsersServices.login(java.lang.String,java.lang.String)

(,)

public abstract java.lang.Boolean com.nemo.webservices.users.NemoUsersServices.updateUser(com.nemo.webservices.users.Nemouser)

()

public abstract java.util.List com.nemo.webservices.users.NemoUsersServices.getWaitingUsers()

()

public abstract java.util.List com.nemo.webservices.users.NemoUsersServices.getAllUsers()

()

Figura 6.10: Métodos do *Web Service* que interage com a base de dados do *Users*.

- **getUser:** Retorna um utilizador indicando o ID pretendido, o nome.
- **login:** Verifica se a *Password* corresponde ao *Username*.
- **updateUser:** Guarda informação actualizada sobre utilizador.
- **getWaitingUsers:** Retorna todos os utilizadores que estão em estado de espera.
- **getAllUsers():** Retorna todos os utilizadores registados na base de dados.

Capítulo 7

Conclusões

Neste sétimo e último capítulo, é feita uma síntese de todo o trabalho feito atendendo aos objectivos a que o trabalho foi proposto. Este capítulo está dividido em três secções: a secção 7.1 onde se faz uma pequena síntese do trabalho desenvolvido; na secção 7.2 são feitas as considerações finais da dissertação e por último, na secção 7.3 são enumerados alguns tópicos para trabalho futuro.

7.1 Síntese Geral do Trabalho

A crise de energia, que se verifica na actualidade, trouxe dois protagonistas que estão interligados: a eficiência energética e as fontes de energia renováveis. Existe portanto, uma crescente necessidade de eficiência quanto ao uso de energia, com vista a evitar o desperdício da mesma e tentar que seja o mais sustentável ou ecologicamente correcto o seu uso. A eficiência energética tornou-se então, neste momento, um dos pilares da sustentabilidade global. Existe por isso, a necessidade de utilizar mecanismos com custos financeiros menores que permitam poupar e gerir os recursos existentes com o uso de energias renováveis, com vista a uma maior eficiência energética.

O trabalho apresentado nesta dissertação, tem como objectivo a criação de uma infra-estrutura de *software* que sirva de apoio à produção, consumo e distribuição de energia. Para possibilitar o cumprimento dos objectivos a que o trabalho foi proposto, a plataforma desenvolvida possui as seguintes características:

- Possibilita a elaboração duma base de dados que possui a informação relevante sobre os diversos IEDs existentes no sistema, bem como a criação de outra que engloba a informação sobre os diferentes serviços que cada um disponibiliza.
- Permite a criação duma base de dados com os diversos utilizadores registados no sistema, para possibilitar que a infra-estrutura possa ser acedida.
- Permite o controlo e monitorização das três bases de dados referidas ao longo desta dissertação.

- Permite o acesso à plataforma desenvolvida através de um portal *Web* que foi desenvolvido, sendo este auxiliado pelos diferentes *Web Services* criados.

7.2 Conclusões

Nesta dissertação foram mostradas inúmeras situações onde a monitorização de dispositivos relacionados com energia é de verdadeira importância, tanto a nível de poupança de energia, como também a nível da melhoria do aproveitamento da mesma e onde as tecnologias de informação são usadas como ferramentas para o efeito.

O NEMO-M apresenta-se como uma contribuição no suporte a um sistema de gestão de energia que necessite de controlar dispositivos nas diversas áreas relacionadas com a energia. Este *software* já apresenta resultados sólidos no sentido de explorar os benefícios potenciais do uso de *standards* como auxílio às plataformas de *software*.

O bom uso das diversas bases de dados, bem como do portal *Web* elaborado, demonstra que através aproveitamento apropriado de todas as suas funcionalidades irá permitir que o sistema possa vir a ser controlado pelo respectivo sistema de gestão de energia.

O uso de diversos *standards*, entre eles o *standard IEC61850* e o XML, possibilita que o NEMO-M seja interoperável. O *standard IEC61850* assegura a facilidade de integração e virtualização de novos IEDs no sistema. O uso do XML como parte integrante na troca de mensagens e ficheiros no *software* e com a criação de ficheiros auxiliares baseados no mesmo para complementar as informações sobre cada dispositivo, possibilita a integração de dispositivos desconhecidos ao sistema, dando assim consistência necessária para o futuro PnP do NEMO-M, tornando-a também escalável.

7.3 Trabalho Futuro

Como já fora referido na secção 1.4, este trabalho foi desenvolvido no âmbito do projecto NEMO que ainda se encontra em desenvolvimento, não permitindo por isso desenvolver, nem testar todas as funcionalidades desejadas, possibilitando por isso que novas funcionalidades venham ainda a ser realizadas.

Assim, como aspectos futuros a serem tratados relacionados com o *software* desenvolvido, o NEMO-M, sugerem-se as seguintes funcionalidades:

- Desenvolvimento dum maior número de *Web Services*, que irão permitir aos utilizadores da infraestrutura uma maior gama de funcionalidades e possibilitará o total controlo e monitorização da plataforma.
- Desenvolvimento do portal *Web*, tendo em vista o uso dos novos serviços que irão ser disponibilizados para o administrador.

- Desenvolvimento das funcionalidades que irão permitir ao utilizador comum da plataforma aceder à mesma, visto este trabalho apenas consistir nas funcionalidades principais de um administrador dum sistema.

Bibliografia

- [WS0, 2004] (2004). Web services architecture ver <http://www.w3.org/tr/ws-arch/>.
- [SQL, 2011] (2011). About mysql ver <http://www.mysql.com/>.
- [XML, 2011] (2011). Extensible markup language (xml) ver <http://www.w3.org/xml/>.
- [INT, 2011] (2011). Interoperability ver <http://searchsoa.techtarget.com/definition/interoperability>.
- [JSP, 2011] (2011). Javaserer pages technology ver <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>.
- [NET, 2011] (2011). Netbeans ide 6.9.1 release information ver <http://netbeans.org/community/releases/69/index.html>.
- [HIB, 2011] (2011). Relational persistence for java and .net ver <http://www.hibernate.org/>.
- [SCA, 2011] (2011). Scada systems ver <http://www.scadasystems.net/>.
- [UML, 2011] (2011). Visual paradigm ver <http://www.visual-paradigm.com/>.
- [Apostolov and Vandiver, 2004] Apostolov, A. and Vandiver, B. (2004). Functional testing of iec 61850 based ieds and systems. In *Power Systems Conference and Exposition, 2004. IEEE PES*, pages 640 – 645 vol.2.
- [Apostolov, 2010] Apostolov, A. P. (2010). Uml and xml use in iec 61850. In *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*, pages 1 –6.
- [Baigent et al., 2003] Baigent, D., Adamiak, M., and Mackiewicz, R. (2003). *IEC 61850 Communication Networks and Systems In Substations: An Overview For Users*. IEC Standard.
- [Barry, 2011] Barry, D. K. (2011). Service-oriented architecture (soa) definition. http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html.
- [Candido et al., 2009] Candido, G., Jammes, F., Barata, J., and Colombo, A. (2009). Generic management services for dpws-enabled devices. In *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*, pages 3931 –3936.
- [Chao et al., 2010] Chao, K.-M., Shah, N., Farmer, R., Matei, A., Chen, D.-Y., Schuster-James, H., and Tedd, R. (2010). A profile based energy management system for domestic electrical appliances. In *e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on*, pages 415 –420.

- [Gasser and Palfrey, 2007] Gasser, U. and Palfrey, J. (2007). Breaking down digital barriers: When and how ict interoperability drives innovation. Berkman Center for Internet and Society, Harvard Law School.
- [Geberslassie and Bitzer, 2010] Geberslassie, M. and Bitzer, B. (2010). Future scada systems for decentralized distribution systems. In *Universities Power Engineering Conference (UPEC), 2010 45th International*, pages 1 –4.
- [Hossenlopp, 2007] Hossenlopp, L. (2007). Engineering perspectives on iec 61850. *Power and Energy Magazine, IEEE*, 5(3):45 –50.
- [Huo et al., 2008] Huo, Z., Zhang, L., and Zhang, Z. (2008). Research on graphics model design for iec61850 scl visual configuration. In *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, volume 2, pages 700 –704.
- [Jing, 2010] Jing, Z. (2010). The analysis of xml technology in network security. In *Intelligence Information Processing and Trusted Computing (IPTC), 2010 International Symposium on*, pages 701 –704.
- [Jogalekar and Woodside, 1998] Jogalekar, P. and Woodside, M. (1998). Evaluating the scalability of distributed systems. In *System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on*, volume 7, pages 524 –531 vol.7.
- [Jogalekar and Woodside, 2000] Jogalekar, P. and Woodside, M. (2000). Evaluating the scalability of distributed systems. *Parallel and Distributed Systems, IEEE Transactions on*, 11(6):589 –603.
- [Kostic and Frei, 2007] Kostic, T. and Frei, C. (2007). Modelling and using iec 61850-7-2 (acsi) as an api. In *Power Tech, 2007 IEEE Lausanne*, pages 713 –719.
- [Lee, 2010] Lee, Y. (2010). Interoperability management for soa by relative proof and test suit. In *Advanced Information Management and Service (IMS), 2010 6th International Conference on*, pages 515 –520.
- [Lima et al., 2011a] Lima, C., Gomes, V., Lima, J., Martins, J. F., Barata, J., Ribeiro, L., and Cândido, G. (2011a). A standard-based software infrastructure to support energy efficiency using renewable energy sources.
- [Lima et al., 2010] Lima, C., Martins, J. F., Barata, J., Ribeiro, L., and Cândido, G. (2010). Towards a service based infrastructure to improve efficiency into energy systems, the nemo coded quest.
- [Lima et al., 2011b] Lima, J., Lima, C., Gomes, V., Martins, J. F., Barata, J., Ribeiro, L., and Cândido, G. (2011b). Dpws as specific communication service mapping for iec 61850.
- [Ma et al., 2010] Ma, X., Cui, R., Sun, Y., Peng, C., and Wu, Z. (2010). Supervisory and energy management system of large public buildings. In *Mechatronics and Automation (ICMA), 2010 International Conference on*, pages 928 –933.
- [MacKenzie et al., 2006] MacKenzie, C. M., Laskey, K., McCabe, F., Brown, P. F., and Metz, R. (2006). Reference model for service oriented architecture 1.0. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm.

- [Mercurio et al., 2009] Mercurio, A., Di Giorgio, A., and Cioci, P. (2009). Open-source implementation of monitoring and controlling services for ems/scada systems by means of web services; iec 61850 and iec 61970 standards. *Power Delivery, IEEE Transactions on*, 24(3):1148 –1153.
- [Microsoft, 2009] Microsoft (2009). *Standards in the ICT Industry*. Microsoft.
- [Ribeiro et al., 2008] Ribeiro, L., Barata, J., and Colombo, A. (2008). Mas and soa: A case study exploring principles and technologies to support self-properties in assembly systems. In *Self-Adaptive and Self-Organizing Systems Workshops, 2008. SASOW 2008. Second IEEE International Conference on*, pages 192 –197.
- [Stum et al., 1997] Stum, K., Mosier, R., and Haasl, T. (1997). *Energy Management Systems-a Practical Guide*. PECL.
- [Xu et al., 2007] Xu, T., Hou, H., Yu, H., You, D., Yin, X., and Wang, Y. (2007). Analysis on iec 61850 interoperability support. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1 –6.